Title
Planning Under Uncertainty for Human-Compatible Robots

Author
Liu, Chang

Publication Date
2017

Peer reviewed|Thesis/dissertation

Planning Under Uncertainty for Human-Compatible Robots

by

Chang Liu

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Engineering – Mechanical Engineering

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Francesco Borrelli, Chair
Professor Masayoshi Tomizuka
Professor J. Karl Hedrick
Professor Thomas Griffiths

Fall 2017

Planning Under Uncertainty for Human-Compatible Robots

Abstract

Planning Under Uncertainty for Human-Compatible Robots

by

Chang Liu

Doctor of Philosophy in Engineering – Mechanical Engineering

University of California, Berkeley

Professor Francesco Borrelli, Chair

Recent progress in robotic systems has significantly advanced robot functional capabilities, including perception, planning, and control. As robots are gaining wider applications in our society, they have started entering our workplace and interacting with us. This leads to new challenges for robots: they are expected to not only be more functionally capable automatic machines, but also become human-compatible, which requires robots to make themselves competent agents to *work for* people and collaborative partners to *work with* people on diverse tasks. The capability to planning under uncertainty lies at the core to achieving this goal. The aim of this dissertation is to develop new approaches that improve the autonomy and intelligence of robots to enable them to reliably work for and with people. Especially, this dissertation investigates uncertainty reduction and the planning under various types of uncertainty with the focus on three related topics, including distributed filtering, informative path planning, and planning for human-robot interaction.

In the first topic, the dissertation studies uncertainty reduction via distributed filtering using networked robots. We consider the distributed version of the generic Bayesian filter. Two new methods of measurement exchange among networked robots are proposed, which enable the dissemination of robots' sensor measurements in time-invariant and time-variant communication networks. By using such methods, the communication burden of the robot network can be significantly reduced compared to traditionally used methods. Based on these measurement exchange methods, we develop two distributed Bayesian filters for time-invariant and time-variant networks. It has been proved that the proposed distributed Bayesian filter can achieve consistent estimation. The application in target localization and tracking is presented.

In the second part, the dissertation focuses on planning under the uncertainty of target position and motion model. This part investigates the path planning of a mobile robot to autonomously search and localize/track a static/moving target. We first study the case of linear Gaussian sensing and mobility models. A path planning approach based on model predictive control (MPC) is proposed, which uses a modified Kalman filter for uncertainty prediction and a sequential planning strategy for path generation. We then investigate the path planning in a dynamic environment, with the sensor using a binary model. A closed-form objective function for the MPC-based path planner is proposed, which significantly reduces the computational complexity. The safety of robot is enforced by using a barrier function in the objective function of MPC.

The first two topics concentrate on making robots autonomously work for people. In the third topic, the dissertation addresses the demands to make robots work with people and achieve coordination. We first consider the planning of robots under the uncertainty of humans' trajectory in a human-following application, where the robot needs to generate a path to follow a person in a safe and comfortable way. We propose a model-based human motion prediction approach using the principle of interacting multiple model estimation. A path planner based on nonlinear MPC is then developed for the robot to generate human-following paths, which takes into account the safety and comfort of the accompanied person.

We then investigate the planning of robots under the uncertainty of humans' internal states, including their intention and belief. Especially, the task planning in the human-robot collaboration is considered. We develop an adaptive task planning scheme that allows a robot to use motion-level inference to understand a human partner's plan and then adjust its task-level plan to coordinate with the person. In addition, we model a person's inference process and develop a task planning approach for a robot to generate human-predictable plans, which aims to reduce the misalignment between people's belief and robots' plan.

To my parents.

Contents

List of Figures

List of Tables

Acknowledgments

I would like to express my sincere gratitude to my dissertation committee members, Prof. Francesco Borrelli, Prof. J. Karl Hedrick, Prof. Masayoshi Tomizuka, and Prof. Thomas Griffiths. I have benefited tremendously from discussing and taking courses with Francesco and Masayoshi, who had guided me into the exciting world of Control. Francesco is also very kind to serve as my dissertation committee chair. I have gained a lot from Tom's instruction and advising, which has led me into the field of computational cognitive science and significantly shaped my research. Besides, they have provided invaluable support on my academic career, which I really appreciate.

I want to express my deepest gratitude to Prof. J. Karl Hedrick, who had advised me for five years since I joined the Vehicle Dynamics & Control lab. I am grateful for him to accept me into his group at the beginning of year 2012 when I switched my major field of study to Control, even though I had no idea what a state space or a Kalman filter was at that time. His trust and confidence in me had completely changed my career path and I am still benefiting from it today. During the past five years working with him, I have learned so much about enjoying research, collaborating with people, and living a worthy life. It was a great sorrow to me when he unfortunately passed away in 2017. I really miss him.

I have been extremely fortunate to have collaborated with Prof. Anca Dragan and Prof. Shengbo Li, having them as my mentors. Anca has greatly improved my research capability, writing style, and presentation skills. More importantly, she has imparted her passion and quest for novel and significant ideas to me. I am very grateful for this experience. I have collaborated with Shengbo for more than two years. His enthusiasm for research and serious attitude to work has driven me to pursue an academic career thus far.

I also would like to express my sincere gratitude to Prof. Liwei Lin, who served as my academic advisor when I came to Berkeley and kindly allowed me to switch my major field later. Liwei is such a nice and responsible advisor. Without his help and support, I would never have been what I am today.

I also want to thank my peer collaborators, Jessica Hamrick, Jaime Fisac, Donghan Lee, Yi-Wen Liao, Shih-Yuan Liu, and Elena Carano. Jess and Jaime, you are such talented researchers and I have always been amazed by your creative ideas and dedication to work. I feel very fortunate to have worked with both of you and I have learned a lot. Donghan, Yi-Wen, Shih-Yuan, and Elena, you are awesome labmates, collaborators, and friends. I have learned much from working with you. I hope we can keep our collaboration in some way in the future.

I would like to thank all my colleagues and comrades at VDL and MPC, who make my Ph.D. life an interesting and joyful journey. I also want to acknowledge my friends over the years, without whom my life at Berkeley would be much less enjoyable: Yumeng Xi, Tianyao Zhou, Xuance Zhou, Minghui Zheng, Yuting Wei, Wenchao Yu, Xiao Ji, Yi-Wen Liao, Chung-Yen Lin, Chen-Yu Chan, Ruoyun Hua, Min Ting.

My final thanks go to my parents, Chuantie Liu and Xilin Zhang, who encouraged me to go abroad and pursue whatever I like. To me, they are not only parents, but also mentors and friends. They have accompanied me through ups and downs. They have given me their unconditional love and support. They respect my decision and help me explore my true desire. I will try my best to make their support worthwhile.

Chapter 1

Introduction

Robotic systems have gained greater autonomy and become more capable than what they were a decade ago, due to progress in techniques of control and artificial intelligence, methodologies of system design, and computational power of hardware. Advances in robots have given birth to a number of innovative products, such as unmanned aerial vehicles, surgical robots, and industrial robots, which have been effective in carrying out arduous and dangerous duties and increasing productivity. In recent years, robots have started entering human workplace and interacting with people, with personal robots and industrial co-robots as notable examples. Working in the presence of humans has raised the needs to develop *human-compatible* robots that are competent assistants to *work for* people and collaborative partners to *work with* people.

The successful operation of robotic systems relies on the integration of several core components (Figure 1.1), including perception (filtering), planning (decision making), and control. Perception enables a robot to understand the environment and its own states. Filtering is the key technique for perception, which uses sensor measurements to gradually reduce uncertainty of environment and system states and form more accurate estimation. Planning and control allow a robot to generate and execute plans to successfully complete different tasks. Since common uncertainties in real applications, including sensing and actuation errors, modeling errors, and unpredictability of operating environments, can only be reduced but not be completely removed, robots' capability of planning under uncertainty is of paramount importance. In addition, once working with people, robots need to deal with some additional uncertainties, such as human's kinematic states, intention, and belief. Because of this, making human-compatible robots requires efficient uncertainty reduction and reliable planning that can handle various sources of uncertainty. There are several challenges to achieve this goal. First, efficient uncertainty reduction using multi-robot systems is a challenging problem. The traditional centralized coordination paradigm that uses a central operation unit for the whole system is becoming less effective because of increasing demands on reliability and scalability. Second, fast and reliable planning of robots under uncertain environment is an open question and the focus of current research efforts. Successfully solving this problem requires robots to effectively complete tasks while ensuring safety and efficiency. Third, as robots have more interaction with people, traditional methods that treat humans as either commanders or mere irrelevant agents without considering their interaction with robots are no longer effective. Incorporating human-robot mutual interaction into robots' planning process is crucial. This

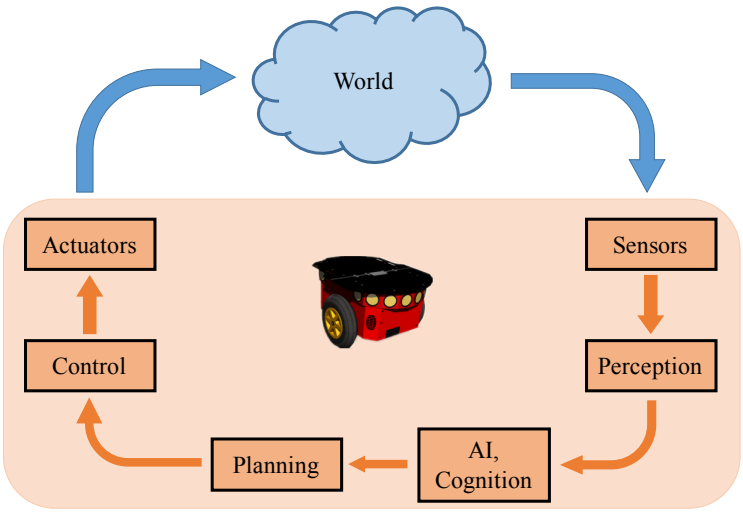


Figure 1.1: An autonomous robotic system is a synergetic combination of perception, planning, and control.

dissertation aims to address these problems in a systematic way, with the focus on planning and estimation of robotic systems.

The overarching goal of the dissertation is to develop the autonomy and intelligence of robots for making effective and reliable decisions under the uncertainties of external states, such as object positions and human trajectories, and the uncertainties of internal states, such as humans' intention and belief. To achieve this goal, the dissertation has proposed new methods utilizing theories and techniques from control, artificial intelligence, optimization, and cognitive science. The dissertation work consists of two main parts. In the first part, we focus on making robots effectively work for people. This part focuses on the informative path planning and distributed filtering using mobile robots, with applications to target search and tracking. In the second part, we focus on making robots collaboratively work with people. This part investigates the interaction between humans and robots, and develops planning approaches for robots to generate behavior that is cognizant of human's plan or belief in order to achieve fluent coordination with people. There are three main topics this dissertation has contributed to, including distributed filtering, informative path planning, and human-robot interaction. An overview of these topics is summarized in Figure 1.2 and listed in following sections.

1.1 Motivation and Overview

1.1.1 Distributed Filtering

Collaborative filtering using a group of networked robots has been widely utilized for estimating environment status [73] due to its merits on low cost, high efficiency, and strong reliability. Example applications include signal source seeking [8], intruder detection [27], and simultaneous

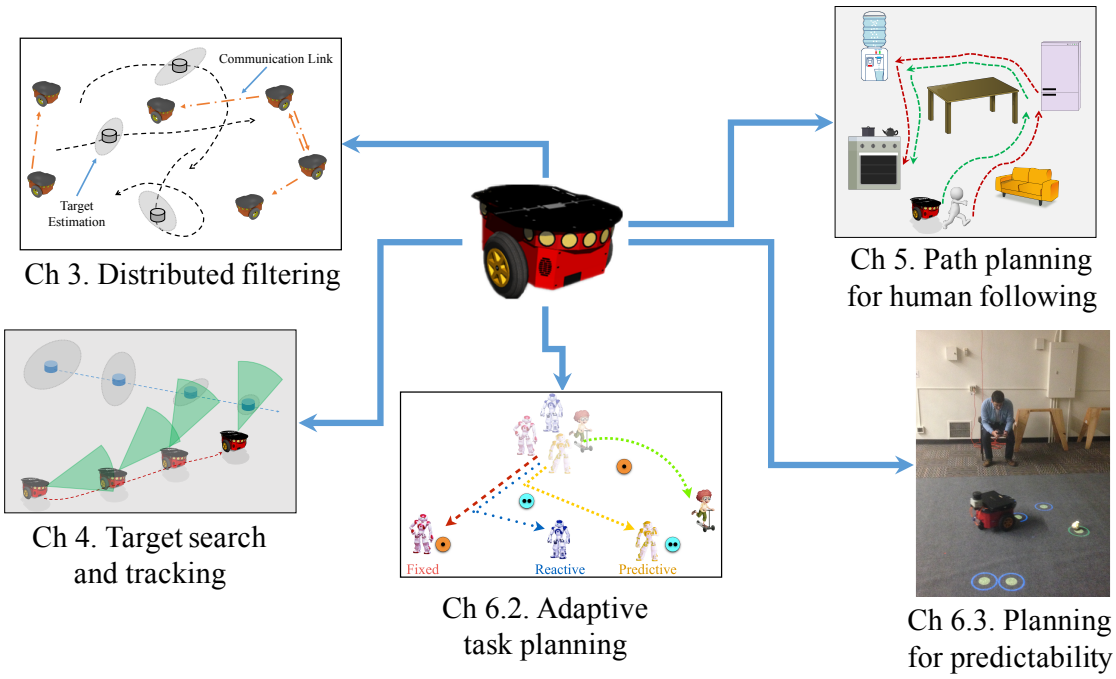


Figure 1.2: An overview of contents in this dissertation.

localization and mapping [37]. The commonly used filtering approaches include the Kalman filter (KF), extended Kalman filter (EKF), Unscented Kalman filter (UKF), particle filter (PF), and Bayesian filter (BF) [158]. The KF is the optimal filter for linear systems with additive Gaussian white noise. EKF and UKF extend KF to nonlinear systems, utilizing the techniques of linearization and unscented transformation, respectively. PF is a type of particle Monte Carlo methodology to solve filtering problems, relying on using samples for approximating the underlying probability distribution. BF is one of the most generic approaches for filtering, which uses Bayes' rule for incorporating measurements. It is applicable to general nonlinear systems [10, 86] and can be reduced to other commonly used filters under certain conditions [32]. For example, with the assumption of linearity and Gaussian noise, a BF can be reduced to a KF; For general nonlinear systems, a BF can be numerically implemented as a PF. Because of this generality, this dissertation investigates the networked variants of BF and develops distributed BFs for tracking targets using only local communication between neighboring robots under various network connectivity conditions.

1.1.2 Informative Path Planning

Research on path planning of robotic systems has a long history, dating back to late 1960's [101]. In recent years, the planning of autonomous robots for information gathering has attracted wide interest and been applied to various applications, such as target search and tracking, persistent monitoring, and environment exploration. The dissertation work investigates the target search and tracking problem. The goal is to utilize an autonomous robot to first find the target of interest and

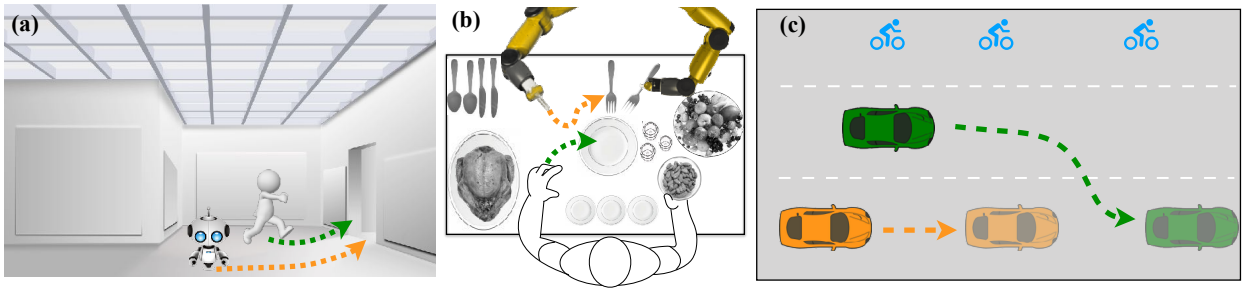


Figure 1.3: Robots can assist and collaborate with humans. (a) A companion robot understands that the elderly person is going to enter a house. So it follows a safe trajectory to open the door for the person. (b) An assistive robot sets up the table with the person. It realizes that the person is going to pick the plate. So the robot changes its target and picks the fork. (c) When the autonomous vehicle (blue car) notices the lane change intention of the human-driven vehicle (green car), it decelerates to give an obvious indicator to the human driver that the autonomous vehicle is yielding the road.

then keep tracking the target afterwards. Some practical applications include indoor target search [30], marine search and rescue [56], and object identification [33].

Studies on search and tracking were initiated during World War II [125]. However, later research efforts have treated these two modes rather separately. On one hand, much work has been focused on tracking moving targets [133, 105, 70], which assumes that targets are initially inside the sensing domain of the target tracker (e.g., a robot). The sensor is then controlled to maintain the target's visibility over time. On the other hand, intense efforts have been put into search problems, the goal of which is to find the targets that are initially of unknown positions. Works on this topic started with simple environment coverage strategies and gradually evolved to more advanced approaches for handling complex, dynamic and stochastic environments [153, 19]. Probabilistic measures, such as probability of detection, and information-theoretic measures, such as entropy, are usually used as metrics for generating search paths. This dissertation work focuses on developing a unified framework for an autonomous robot to search for and track a target under the uncertainty of target position and motion model, while considering different types of sensors and targets, limited sensing domain, and the safety of robots.

1.1.3 Human-Robot Interaction

Once confined to controlled environments devoid of human individuals, robots today are entering our homes and workplaces by unfolding advances in autonomy. From sharing workcells [106], to assisting elderly people [93, 171, 14], to helping in surveillance [65] and search and rescue [56, 159, 110], robots have started operating in human environments, both with and around people.

While autonomous systems are becoming less reliant on human oversight, it is often still desirable for robots to assist and collaborate with human partners in a peer-to-peer structure [53, 39, 150]. Collaboration hinges on coordination and the robot's ability to infer a human's intention or belief and adapt [160, 168]. For example, in the scenario in Figure 1.3a, where a companion robot walks with a person inside a room, the robot is expected to follow the person in a safe and

comfortable way. This requires the robot to plan its path under the uncertainty of person's future trajectory while ensuring safety and comfort. In the collaborative manipulation scenario in Figure 1.3b, where both the human and the robot work together to set up a table, it is necessary for the robot to infer the object that the human will take so that it can avoid hindering or even injuring the human. This requires the robot to infer the person's goal and adjust its own plan to coordinate.

Moreover, achieving seamless team collaboration also requires team members to mutually understand the intent and anticipate needs of other teammates [179]. This means it is necessary for the robot to make plans that are easy for a person to predict to avoid the so-called "automation surprises" [145]. We take the transportation scenario in Figure 1.3c for example. After detecting the human driver's lane change intention, the blue autonomous vehicle can either maintain its current speed or decelerate to allow the green human-driven car to change the lane. However, the deceleration of the blue car is a strong indicator of its plan to yield the road while keeping the speed may be perceived by the human driver as no intention to yield. To avoid such ambiguity, the autonomous vehicle needs to take into account the human drivers' inference process to accordingly generate plans that the driver can easily understand.

Driven by these needs, the dissertation aims to develop planning approaches that achieve following three goals: (1) The robot accurately predicts short-term human trajectory and generates safe and comfortable human-following behavior. (2) The robot infers the human partner's intention and plan adaptively to coordinate. (3) The robot explicitly plans for predictability so that a human partner can confidently predict the robot's plan along the way.

1.2 Outline and Contributions

Chapter 2 reviews relevant concepts and techniques used in the dissertation. Chapter 3 proposes a distributed Bayesian filter approach for networked robots in the presence of time-invariant and time-varying network topologies. Chapter 4 presents an informative path planning approach based on model predictive control for a mobile robot to search and localize/track a static/moving target. This chapter first considers the linear sensor model and target motion model. The case of binary sensor model is subsequently investigated. Chapter 5 and Chapter 6 focus on human-robot interaction. Chapter 5 proposes a human motion prediction method and a path planning approach for a companion robot to generate human-following trajectories under the uncertainty of human's future trajectory. Chapter 6 subsequently investigates planning for human-robot collaboration with the focus on task planning under the uncertainty of human's internal states, including intention and belief. Specifically, this chapter first presents an adaptive planning scheme for a robot to infer its human partner's plan and then coordinate with the person to jointly complete a set of tasks. It then develops a task planning method for a robot to generate human-predictable plans, which enables a human partner to confidently infer the remaining part of the plan after observing the initial part of it. Chapter 7 summarizes the main contributions in the dissertation and provides ideas for future research. Below is a brief overview of contributions in each chapter with respect to the goals listed above.

Chapter 3 We propose two methods of information sharing among networked robots, the *Latest-In-and-Full-Out* (LIFO) method and the *Full-In-and-Full-Out* (FIFO) method. The former is developed for time-invariant communication networks and the latter is for time-varying communication networks. We show that, by using such methods, the communication burden of each robot is either linear or quadratic of the network size (the number of robots), depending on the variability of the network over time. Based on these information sharing methods, we have developed two distributed Bayesian filters, which forms a generic nonlinear filtering framework for distributed filtering in the presence of fixed and dynamically changing networks, respectively. It is proved that the proposed distributed Bayesian filter can achieve consistent estimation, guaranteeing that each robot can correctly estimate unknown states asymptotically.

Chapter 4 We propose a trajectory-optimization-based path planning approach for target search and tracking. Under the condition that the target motion and sensor measurement models are linear Gaussian, we develop a sequential planning approach using a modified Kalman filter and the nonlinear model predictive control for the robot to effectively search for and track a moving target. We then consider using a binary sensor to localize a static target in a dynamic environment that contains both static and moving obstacles. We derive a closed-form objective function for the path planner, which significantly reduces the computational complexity. The safety of robots is enforced by adding a barrier function in the objective function.

Chapter 5 In this chapter, we propose a path planning approach to generate human-following trajectories in cluttered environments for a mobile robot. A model-based human motion prediction approach, called the Parallel Interacting Multiple Model-Unscented Kalman Filter, is proposed to predict short-term human trajectory. We then develop a path planner based on nonlinear model predictive control for generating human-following trajectory, which takes into account the safety and comfort considerations.

Chapter 6 We propose two task planning frameworks for human-robot collaboration in this chapter. The first one focuses on enabling robots to infer humans' plan and then adapt for coordination. This work combines the motion-level goal inference with the task-level adaptive planning to allow an assistive robot to help a human partner in completing a set of heterogeneous tasks. In the second framework, we develop a model of people's inference about robots' plan and enable a robot to plan explicitly for the predictability of its plan. To achieve this, we propose a new metric, called the t-predictability, to evaluate the predictability of a robot's plan. A t-predictable planner is subsequently developed for a robot to generate human-predictable plans so that people can confidently infer the robot's plan along the way.

Chapter 2

Background and Related Work

2.1 System Dynamics and Sensor Modeling

A system dynamic model can be described as a differential equation and a sensor measurement model can be represented as an algebraic equation. This type of representation clearly describes the temporal change of system states and is widely used in the Control field [88]. Another representation uses two conditional probabilities to describe system evolution, which is commonly used in the Robotics field [158]. This section gives an overview of both representations.

2.1.1 System Dynamics

The evolution of a system can be described as a dynamic system using difference equation (for discrete-time systems):

$$x_{k+1} = f(x_k, u_k) + v_k, \tag{2.1}$$

where $x_k \in \mathbb{R}^n$ represents the system state at time k, $u_k \in \mathbb{R}^p$ is the control input of the system, and $v_k \in \mathbb{R}^n$ is the additive process noise[1]. The conditional probability representation of system dynamics, called the *state transition probability*, can be defined as $P(x_{k+1}|x_k, u_k)$. The distribution of $P(x_{k+1}|x_k, u_k)$ relies on f and distribution of v_k.

For example, a stochastic linear time-invariant system with additive zero mean Gaussian noise can be represented as

$$x_{k+1} = Ax_k + Bu_k + v_k,\ v_k \sim \mathcal{N}(0, Q), \tag{2.2}$$

where $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times p}$ are system matrices; $v_k \in \mathbb{R}^n$ is a zero-mean Gaussian noise with $Q \geq 0$ being the covariance matrix. Then $x_{k+1} \sim \mathcal{N}(Ax_k + Bu_k, Q)$. The conditional probability representation $P(x_{k+1}|x_k, u_k)$ can be computed using this Gaussian distribution.

Notice that the system dynamic model defined here is a general formulation. This dissertation will use this formulation to represent robot kinematic model, target motion model, and human

[1] In general, the noise is not necessarily additive. However, in most systems in reality and all the systems in this dissertation assumes the additive noise. Therefore we use this kind of noise in the general formulation of the model with the hope that this will not confuse readers.

motion model, where f^r, f^g, f^h and x^r, x^g, x^h are used to represent corresponding functions and states.

2.1.2 Sensor Modeling

The sensor model can be represented as an algebraic equation:

$$z_k = h(x_k^g; x_k^s) + w_k, \tag{2.3}$$

where $z_k \in \mathbb{R}^q$ is the sensor measurement at time k. $x_k^g \in \mathbb{R}^n$ represents the state of a target that is measured by the sensor. $x_k^s \in \mathbb{R}^m$ is the sensor state[2] (e.g., position, orientation) and $w_k \in \mathbb{R}^q$ is the measurement noise. The measurement function h depends on the type of the sensor. Several typical sensors are listed as follows [16]:

Range-only sensors h is a function of the relative Euclidean distance between the sensor and the target:

$$h(x_k^g; x_k^s) = \|x_k^g - x_k^s\|_2,$$

where $\|\cdot\|_2$ is the Euclidean norm in S.

Bearing-only sensors h is a function of the relative bearing between the sensor and the target:

$$h(x_k^g; x_k^s) = \measuredangle(x_k^g - x_k^s),$$

where $\measuredangle$ denotes the angle of a vector.

Range-bearing sensors h includes both the relative distance and bearing:

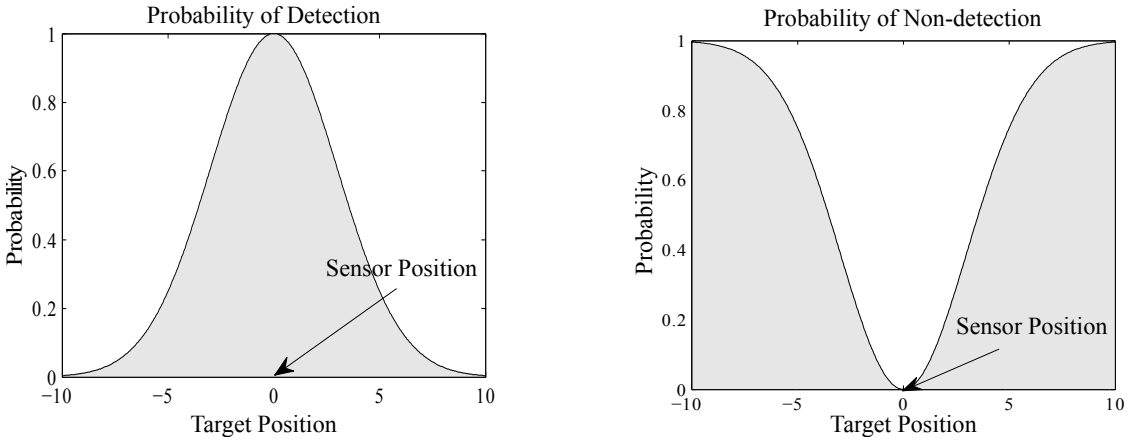
$$h(x_k^g, x_k^s) = x_k^g - x_k^s.$$

The conditional probability representation of sensor measurement model $P(z_k|x_k^g; x_k^s)$, known as the *measurement probability*, relates the sensor measurement as a random variable of the underlying states. Its distribution is determined by the sensor type and distribution of measurement noise. For example, a linear measurement model with additive zero mean Gaussian noise can be represented as

$$z_k = Cx_k^g + w_k, \;\; w_k \sim \mathcal{N}(0, R). \tag{2.4}$$

The corresponding measurement probability model is $z_k \sim \mathcal{N}(Cx_k^g, R)$ and $P(z_k|x_k^g; x_k^s)$ can be computed using this Gaussian distribution.

[2] In this section, we use x^s to represent the sensor state for generality. In following chapters, the sensor state x^s is considered interchangeable with the robot state x^r.



(a) Probability of detection.

(b) Probability of non-detection.

Figure 2.1: Probability of detection/non-detection of a binary sensor model.

Binary sensors In many applications, binary sensors are popular due to its low cost. The representation of binary sensors is different from Eq. (2.3). Instead, the sensor only gives two types of observation: **1** (the target is detected) and **0** (no target is detected). The binary sensor can be modeled as

$$z_k = \mathbb{1}_{\{h(x_k^g, x_k^s) \geq \gamma\}}$$

where γ is the threshold that determines whether a target is considered as being detected or not.

There exist several different models for binary sensors, including Gaussian function [17] and step function [41]. For example, assume that the detection only depends on the distance between the sensor and the target, then the probability of detection can be represented as a Gaussian function as follows:

$$P(z_k = \mathbf{1} | x_k^g, x_k^s; \Sigma) = \exp^{-\frac{1}{2}(x^g - x^s)^T \Sigma^{-1} (x^g - x^s)}, \tag{2.5}$$

where Σ is a positive definite covariance matrix. Correspondingly, the probability of non-detection is

$$P(z_k = \mathbf{0} | x_k^g, x_k^s; \Sigma) = 1 - P(z_k = \mathbf{1} | x_k^g, x_k^s; \Sigma). \tag{2.6}$$

Figures 2.1a and 2.1b show the probability of detection and non-detection in one-dimensional case, respectively.

2.2 Filtering Algorithms

Filtering, also known as estimation, is the technique to infer the underlying, indirectly measurable states of a system from measurements related to these states. Figure 2.2 illustrates the paradigm of filtering. Following sections review some commonly used filtering algorithms.

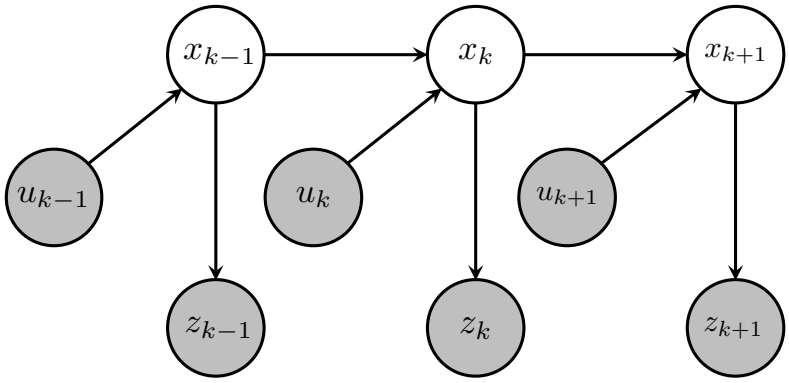


Figure 2.2: The dynamic Bayes network that characterizes the controls, states, and measurements. The gray circles are known states (modified from [158]).

2.2.1 Kalman Filter

Kalman filter is one of the most prominent filtering algorithms, invented by Rudolf Kalman in his seminal paper [90] in 1960. Kalman filter is the optimal filter for linear dynamic systems with Gaussian process and measurement noise. It is also the optimal linear filter for general dynamic systems in the sense of minimum variance estimation.

Suppose the linear system is defined as

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + v_k,\ v_t \sim \mathcal{N}(0, Q), \\ z_k &= Cx_k + w_k,\ w_k \sim \mathcal{N}(0, R), \end{aligned}$$

The Kalman filter consists of two steps:

Prediction

$$\begin{aligned} \hat{x}_{k+1|k} &= A\hat{x}_{k|k} + Bu_k, \\ P_{k+1|k} &= AP_{k|k}A' + Q. \end{aligned}$$

Updating

$$\begin{aligned} K_{k+1} &= P_{k+1|k}C(CP_{k+1|k}C' + R)^{-1}, \\ \hat{x}_{k+1|k+1} &= \hat{x}_{k+1|k} + \gamma_{k+1}K_{k+1}(z_{k+1} - C\hat{x}_{k+1|k}), \\ P_{k+1|k+1} &= P_{k+1|k} - \gamma_{k+1}K_{k+1}CP_{k+1|k}. \end{aligned}$$

The Kalman filter cannot be directly applied for nonlinear systems. Several variants have been developed to extend Kalman filter to nonlinear systems. The most notable one is Extended Kalman Filter (EKF). The EKF uses the 1^{st}-order approximation to linearize both the system dynamics and sensor model by using Taylor expansion and then apply Kalman filter to the linearized system. Its strength lies in its simplicity and computational efficiency. However, EKF is limited by the use of

linear approximation, the goodness of which depends on the degree of uncertainty and the degree of local non-linearity of the functions being approximated [158]. Another important extension of Kalman filter is unscented Kalman Filter (UKF), which will is described in the following section.

2.2.2 Unscented Kalman Filter

The unscented Kalman Filter is another way of linearizing the transformation of a Gaussian, which performs a stochastic linearization via using weighted linear regression [158]. The UKF first deterministically extracts the so-called *sigma points* from the Gaussian distribution of states. These points then undergo the *unscented transformation* and are used to form a new estimate of states, incorporating sensor measurements. Since these sample points completely captures the mean and covariance of Gaussian, the posterior mean and covariance are accurately captured to the 3$^{\text{rd}}$-order (Taylor expansion) [172].

To be specific, given an arbitrary nonlinear motion model $y = g(x)$ and an L-dimensional Gaussian Random Vector (GRV) x with mean $\hat{x}$ and covariance P_x, the statistics of y can be approximated by using $2L + 1$ sigma points $\left\{\chi^{(i)}\right\}_{i=0}^{2L} = \left\{\hat{x} \text{ and } \hat{x} \pm \sigma_j, j = 1, ..., L\right\}$ where σ_j is the j^{th} column of the matrix $\sqrt{(L+\lambda)P_x}$. $\lambda = \alpha^2(L + \kappa) - L$ is a scaling parameter [83].

Once the sigma points have been generated, each point is passed through the nonlinear function $y = g(x)$, i.e. each column of the sigma points is propagated through the nonlinear transformation, as in $\zeta^{(i)} = g(\chi^{(i)}), i = 0, ..., 2L$. The mean $\hat{y}$ and the covariance P_y are approximated as $\hat{y} \simeq \Sigma_{i=0}^{2L} W_i^{(m)} \zeta^{(i)}$ and $P_y \simeq \Sigma_{i=0}^{2L} W_i^{(c)} (\zeta^{(i)} - \hat{z})(\zeta^{(i)} - \hat{z})^{\mathsf{T}}$, with weights and parameters calculated in Eq. (2.7):

$$W_0^{(m)} = \frac{\lambda}{L + \lambda} \tag{2.7a}$$

$$W_0^{(c)} = \frac{\lambda}{L + \lambda} + 1 - \alpha^2 + \beta \tag{2.7b}$$

$$W_i^{(m)} = W_i^{(c)} = \frac{1}{L + \lambda}, \quad i = 1, ..., 2L, \tag{2.7c}$$

where α determines the spread of sigma points about the mean $\hat{x}$; κ is a secondary scaling parameter; β is used to incorporate prior knowledge of the distribution. Readers can refer to [71] and [87] for more details of the UKF algorithm.

2.2.3 Particle Filter

Despite the simplicity of EKF and UKF, these filters can lead to inaccurate estimation when the underlying state distribution significantly deviates from a Gaussian distribution. Particle filter (PF) is a widely used nonparametric nonlinear filter that is suitable for arbitrary distribution of states. The PF approximates the state distribution by a finite number of particles that are sampled from the underlying distribution. Particle filter has been widely used in robotics and adopted for state estimation, localization, and mapping. Algorithm 1 [158] shows the algorithm of *sequential important resampling (SIR) particle filter*, which is an important and commonly used version of PF. Here

$\chi_k = \left\{x_k^1, \ldots, x_k^M\right\}$ is the set of particles sampled from $p(x_k|z_{1:k})$. Readers interested in properties and practical considerations of PF should refer to [158].

Algorithm 1 SIR particle filter (χ_{k-1}, u_k, z_k)

$\bar{\chi}_k = \chi_k = \emptyset$
for $i = 1$ to M **do**
  sample $x_k^i \sim p(x_k|u_k, x_{k-1}^i)$
  $w_k^i = p(z_k|x_k^i)$
  $\bar{\chi}_k = \bar{\chi}_k + < x_k^i, w_k^i >$
end for
for $i = 1$ to M **do**
  draw i with probability proportional to w_k^i
  add x_k^i to χ_k
end for
return χ_k

2.2.4 Bayesian Filter

Bayesian filter (BF) is a generic nonparametric filtering framework. It is usually implemented as a PF or a histogram filter [158]. Under the condition of linear sensor and target models, a BF can be simplified to be a KF [32]. BF directly updates the full distribution (probability density function (PDF)) using sensor measurements. Similar to KF, the BF consists of two steps:

Prediction At time k, prior PDF $P(x_{k-1}|z_{1:k-1})$ is first predicted forward by using the Chapman-Kolmogorov equation:

$$P(x_k|z_{1:k-1}) = \int_{x_{k-1} \in S} P(x_k|x_{k-1}) P(x_{k-1}|z_{1:k-1}) dx_{k-1}, \tag{2.8}$$

where $P(x_k|x_{k-1})$ represents the state transition probability of the target and S is the space of system state.

Updating The PDF is then updated by the Bayes' rule using the newly received measurement at time k, i.e., z_k:

$$P(x_k|z_{1:k}) = KP(x_k|z_{1:k-1}) P(z_k|x_k), \tag{2.9}$$

where $P(z_k|x_k)$ is the measurement probability and K is a normalization factor, given by:

$$K = \left[\int_{x_k \in S} P(x_k|z_{1:k-1}) P(z_k|x_k) dx_k \right]^{-1}.$$

Here we have utilized the commonly adopted assumption in filtering literature [56, 66, 151] that the sensor measurement at current time is conditionally independent from previous measurements given the current state of the target and sensor. This assumption allows us to simplify $P(z_k|x_k, z_{1:k-1})$, which would have appeared in Eq. (2.9), to be $P(z_k|x_k)$.

2.2.5 Interacting Multiple Model Filter

The Interacting Multiple Model (IMM) approach is an estimation algorithm that simultaneously considers several dynamic models as possible underlying dynamics. It assumes that the system obeys one of a finite number of models at each time instance and can undergo model switching over time. The IMM approach has been considered as a mainstream method for estimating maneuvering targets. The IMM utilizes a bank of r number of filters, each corresponding to a different dynamic model. State estimate at time k is computed as a weighted sum of estimate from each filter, as shown in the following formula:

$$\hat{x}_{k|k} = \sum_{j=1}^{r} \mu_{j,k} \hat{x}^j_{k|k}, \tag{2.10}$$

where $\hat{x}^j_{k|k}$ represents the state estimate from the j^{th} filter and $\hat{x}_{k|k}$ is the overall state estimate. $\mu_{j,k}$ stands for the j^{th} mode probability and can be recursively computed as follows:

$$\mu_{j,k} = \frac{1}{c} \sum_{i=1}^{r} L_{ij,k} p_{ij} \mu_{j,k-1},$$

where $L_{ij,k}$ stands for the Gaussian likelihood of receiving the current measurement z_k given all previous measurements $z_{1:k}$ and that the j^{th} model is in effect at time k; p_{ij} represents the mode transition probability from the i^{th} to the j^{th} model; c denotes the normalizing factor. Each filter uses the mixed initial state estimate and covariance from interaction of the r filters, which consists of the combination of the estimates with the mixing probability at previous time step. Readers interested in the details of the IMM approach can refer to [178].

2.3 Motion Planning of Robotic Systems

This section first reviews common kinematic models of robots. Then several classes of motion planning techniques and related works are presented

2.3.1 Motion Models of Mobile Robots

Models of robotic systems play a fundamental role in robot motion planning. Robot models vary widely depending on the types of robots, including mobile robots, biped humanoids, unmanned aerial/underwater vehicles, robot arms, robot hands, and soft robots. The dissertation work mainly focuses on mobile wheeled robots, and therefore a brief review of common models of mobile robots is given in this section. Readers interested in detailed description of these models as well

as general modeling approaches can refer to [102]. Here we define x_k^r and u_k^r as the robot state and input at time k. In consistency with Eq. (2.1), the models are defined as discrete-time systems and is represented as

$$x_{k+1}^r = f^r(x_k^r, u_k^r). \tag{2.11}$$

These models can usually be obtained by using forward Euler discretization. The sampling time is denoted as Δt throughout the dissertation. Note that the element in vectors x_k^r and u_k^r can vary depending on the models.

Simple Car

A *simple car* model, as shown in Figure 2.3a, contains five degrees of freedom[3], including the car's xy-coordinates in the two-dimensional space, heading angle, speed, and front steering angle, i.e., $x_k^r = [x_{1,k}^r, x_{2,k}^r, \theta_k^r, \mu_k^r, \phi_k^r]^T \in \mathbb{R}^5$. Let the distance between the front and rear axles be L, then the simple car model can be represented as

$$f^r(x_k^r, u_k^r) = x_k^r + \begin{bmatrix} \mu_k^r \cos\theta_k^r \\ \mu_k^r \sin\theta_k^r \\ \frac{\mu_k^r}{L} \tan\phi_k^r \\ u_{1,k}^r \\ u_{2,k}^r \end{bmatrix} \Delta t.$$

The control input $u_k^r = \left[u_{1,k}^r, u_{2,k}^r\right]$ include the linear acceleration of the car body and angular acceleration of the steering angle.

Differential Drive

Most indoor mobile robots have two main wheels, each of which is attached to its own motor. A third wheel is placed in the rear to passively roll along. Such type of robots uses *differential drive*, as shown in Figure 2.3b. Their motion can be represented as

$$f^r(x_k^r, u_k^r) = x_k^r + \begin{bmatrix} \frac{r}{2}(u_{l,k}^r + u_{r,k}^r)\cos\theta_k^r \\ \frac{r}{2}(u_{l,k}^r + u_{r,k}^r)\sin\theta_k^r \\ \frac{r}{L}(u_{r,k}^r - u_{l,k}^r) \end{bmatrix} \Delta t,$$

where r is the wheel radius and L is the distance between two wheels. The state of the robot, $x_k^r = [x_{1,k}^r, x_{2,k}^r, \theta_k^r]^T \in \mathbb{R}^3$, consists of the coordinates and heading angle of the robot. Direct control of two angular wheel speed is usually assumed, i.e., $u_k^r = \left[u_{l,k}^r, u_{r,k}^r\right]$.

Unicycle

A unicycle model ignores the distance between the front and back wheels in the simple car model. The state of a unicycle consists of the xy-coordinates, heading angle, and the velocity, i.e., $x_k^r =$

[3]There exist several variants of the simple car model. Here only one version is listed.

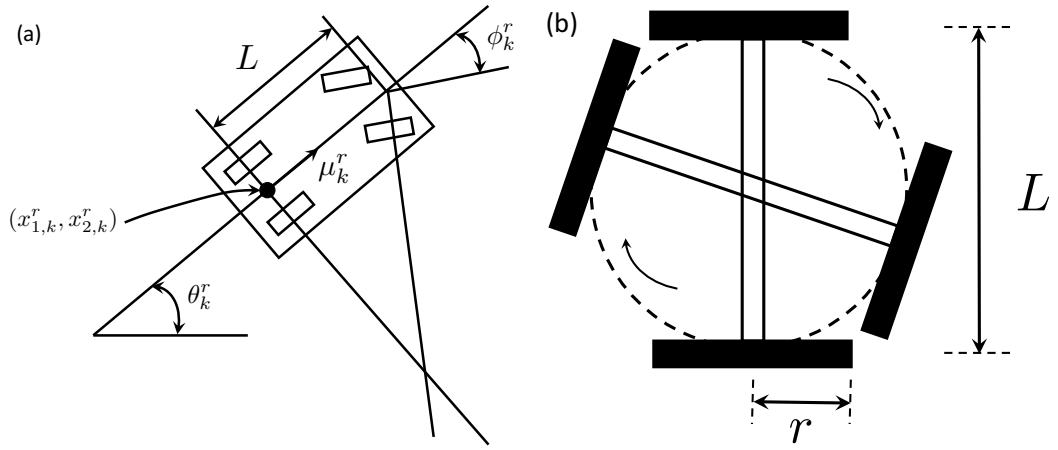


Figure 2.3: (a) The simple car model with five degrees of freedom (modified from [102]). (b) A differential drive car can make pure rotation when the wheels move at opposite velocities(modified from [102]).

$[x_{1,k}^r, x_{2,k}^r, \theta_k^r, \mu_k^r]^T \in \mathbb{R}^4$. Control input $u_k^r = \left[u_{1,k}^r, u_{2,k}^r\right]$ includes the angular velocity and linear acceleration. The unicycle model can be defined as

$$f^r(x_k^r, u_k^r) = x_k^r + \begin{bmatrix} \mu_k^r \cos\theta_k^r \\ \mu_k^r \sin\theta_k^r \\ u_{1,k}^r \\ u_{2,k}^r \end{bmatrix} \Delta t. \tag{2.12}$$

Note that, in some cases, a simple unicycle model with only three states can be used to model the robot's motion, which assumes the direct control on robot speed. This model is then defined as

$$f^r(x_k^r, u_k^r) = x_k^r + \begin{bmatrix} u_{1,k}^r \cos\theta_k^r \\ u_{1,k}^r \sin\theta_k^r \\ u_{2,k}^r \end{bmatrix} \Delta t, \tag{2.13}$$

with the state $x_k^r = [x_{1,k}^r, x_{2,k}^r, \theta_k^r]^T \in \mathbb{R}^3$. The input $u_k^r = \left[u_{1,k}^r, u_{2,k}^r\right]$ now includes the speed and angular velocity.

2.3.2 Motion planning

Motion planning is the fundamental capability of virtually any mechanical systems, including robotic, vehicular, aeronautical, and marine systems. Works on robot motion planning are mainly focused on functional motion such as achieving the minimum travel distance from an initial configuration to the final one while obeying the kinodynamic constraints of a robot. Many fields, such as robotics, artificial intelligence, control theory, and computer graphics have contributed to research on motion planning. The motion planning can be defined either in a continuous space

or a discrete space or even a mixture of these two spaces. Various techniques have been developed based on specific assumption of space. This section will review some mainstream classes of planning methods, including the search-based, curve-interpolation-based, sampling-based, and optimization-based approaches. The model predictive control (MPC) technique, though originally from the optimal control community, has gained popularity in recent years for robot motion planning. Therefore, MPC with its application to motion planning will be described in this section as well.

Search-based and Curve-interpolation-based

Early works on motion planning mainly use graph-search-based planners for efficiently generating shortest collision-free paths on the graph constructed by the discretization of the workspace. Well-known methods such as the Dijkstras algorithm, A* algorithm (and its variants), and the state lattices have all been utilized for real-time planning [89, 164, 47]. In spite of the efficiency and optimality guarantee of these methods, the quality of the planned trajectories highly depends on the resolution of the graph. Besides, the kinematic and dynamic constraints of robots cannot be easily taken into account during the planning process. Curve interpolation planners, including the Clothoid curves [20], polynomial curves [135], spline curves [137] and Bezier curves [139], have also been widely used for online path planning. Similar to the graph search based approaches, the advantage of curve interpolation planners is their low computational cost, since the curve behavior can be defined by only a few control points or parameters. However, the optimality of the generated trajectory cannot be guaranteed. In addition, since the robot kinematics/dynamics is not considered in the planning process, a smoothing process is usually needed to refine the generated paths so that they are kinematically/dynamically feasible for vehicles.

Sampling-based

Sampling-based methods, such as the Probabilistic Road Maps (PRM) and the Rapidly-exploring Random Trees (RRT) [102], have been widely used for robot motion planning since their invention. They offer the benefits of generating feasible trajectories for both holonomic and non-holonomic systems. The RRT* [91] can even guarantee asymptotic optimality of the generated path, which amounts to almost-sure convergence to global optimal solutions with increasing number of samples. These sampling-based methods and their variants have been shown effective in the planning of autonomous vehicles [138, 98], underwater vehicles [82], and unmanned aerial vehicles [72]. However, the practical use of these approaches is hindered by the high computational complexity of the sampling procedure.

Optimization-based

Trajectory-optimization-based approaches have become increasingly popular for robot motion planning recently [185, 117, 112, 147]. The core idea is to formulate the path planning as an optimization problem, which takes into account the desired performance and relevant constraints. The main advantages of these approaches is the flexibility and easiness of encoding all kinds of

requirements on the desired paths of robots. In addition, recent progresses in online optimization solvers, such as the CVX [64] and Ipopt [170] have enabled fast and reliable path generation in real time. Traditionally, trajectory-optimization-based approaches solve a complete trajectory that starts from the starting configuration to the goal configuration. However, since the robot's working environment can usually be dynamic and stochastic, and cannot be fully predicted a-priori, the Model Predictive Control (MPC) approach is commonly used for online path planning of robots in recent years [59, 11, 25]. MPC solves a sequence of finite-time trajectory optimization problem in a recursive manner and therefore can take into account the updating of the environment states during its planning process.

Model Predictive Control

MPC is an advanced optimal control approach to iteratively generate control inputs for constrained dynamic systems. At each time step, MPC computes the optimal control inputs by solving a finite-horizon constrained optimal control problem (FHCOCP), taking into account the system evolution in the planning horizon; Only the control input for the current time step is implemented to the system. At the next time step, FHCOCP is solved again based on the updated system state. A generic formulation of MPC is

$$\min_{u_{k:k+N-1}} \; J(x_{k:k+N}, u_{k:k+N-1}) \tag{2.14a}$$

$$\text{s.t.} \;\; x_{k+i+1} = f(x_{k+i}, u_{k+i}), \tag{2.14b}$$

$$x_{k+i+1} \in \mathcal{X},\; u_{k+i} \in \mathcal{U},\; i = 0, \ldots, N-1, \tag{2.14c}$$

$$x_{k+N} \in \mathcal{X}_f,\; x_k = x(k), \tag{2.14d}$$

where N is the planning horizon and J is the objective function to be minimized; f in Eq. (2.14b) represents the system dynamics; Eq. (2.14c) confines the system state and input to be within the feasible sets $\mathcal{X}$ and $\mathcal{U}$; Eq. (2.14d) is the boundary condition on the terminal state and initial state of the system.

2.3.3 Informative Path Planning

Practical applications of robotic systems have required robots to go beyond merely planning a shortest and collision-free path from a starting configuration to a destination. Increasing demands of using automated systems for search and rescue, environment monitoring, underwater exploration, and military surveillance and reconnaissance necessitate the development of planning methods for information gathering. Planing for information gathering is usually known as informative path planning ot information-driven path planning. The goal is to utilize a single robot or a team of robots to collect certain information in an environment. Several relevant topics have been investigated in robotics, including coverage, persistent monitoring, object detection, and active perception. Among them, target search and tracking has received considerable interest due to its direct applications in environment monitoring and disaster response.

A pioneering work that considered the target search and tracking under the same framework was conducted by Bourgault et al. [18], who designed a unified objective function representing the

cumulative probability of detection for search and tracking. It utilized a Bayesian filter for updating the target's state based on sensor measurements. Due to the numerical complexity of evaluating the objective function, the work used one-step look-ahead scheme for path planning. The idea of using such greedy policy for path planning has also been utilized in [148], where robots follow the gradient of mutual information to estimate environment events and hazards online. Research has also focused on distributed information gathering and target search using the greedy path planning policy [80, 86].

Approaches for non-myopic target search and tracking have also been developed. For example, Tisdale et al. [159] utilized a receding horizon path planning approach for a team of aerial vehicles to cooperatively search for targets, with the probability of detection being maximized over a finite horizon. Ryan et al. [144] used an information-theoretic objective function and developed a control-based formulation to minimize the entropy of the estimated state of a moving target over a multiple-step horizon. The prediction of conditional entropy is computed by sequential Monte Carlo method in the context of particle filtering. However, due to the complexity of Monte Carlo method, such approach could not be implemented in real time. Lanillos et al. [99] used the cross entropy approach to solve an optimization problem with a discounted probability of detection as the objective function, which leads to minimum-time search of the target.

Sampling-based path planning approaches have also been adopted for information gathering. For example, Hollinger et al. [82] has proposed a rapidly exploring information gathering graph approach to effectively optimize a modular or submodular objective function for information gathering. Levine [108] proposed a variant of RRT [103] algorithm to search for targets in the environment, considering the limited sensing domain and occlusion caused by obstacles. However, sampling-based methods suffer from poor scalability to higher dimensional space and thus is of limited use.

Several metrics have been utilized to quantify the information gathering and guide the planning of robots. The table below lists these metrics and literature that use them.

Table 2.1: Common information metrics used for informative path planning

Metrics	Definition	Literature
Probability of detection	$J = 1 - P(Z = \mathbf{0}\|X)$	[159, 57, 99, 173, 114]
Entropy	$H(X) = -\int_X p(X) \log p(X) dx$	[54, 144, 111]
Entropy reduction	$\Delta H(X; Z_j, Z_i) = H(X\|Z_i) - H(X\|Z_i, Z_j)$	[182, 81]
Mutual information	$I(Z; X) = \int_{X,Z} p(X, Z) \log \frac{p(X,Z)}{p(X)p(Z)} dX dZ$	[80, 86, 31, 8]
Renyi entropy	$H_\alpha(X) = \frac{1}{1-\alpha} \log_2 \left(\sum_{i=1}^{n} p_i^\alpha \right)$	[95, 24]
Fisher information	$J = E_X \{ -\nabla_X [\nabla_X \log P(X, Z)] \}$	[162, 108, 124]

2.4 Human-Robot Interaction

The popularity of robots have led to increasing interests in HRI. Traditionally, the interaction between humans and robots are investigated from the perspective of behavioral analysis, which studies the implications that actions, facial expressions, and gazes convey. In recent years, the confluence of Artificial Intelligence, Control, and Psychology has led to the so-called *algorithmic human-robot interaction* that systematically investigates HRI using mathematical tools such as inference, planning, and optimization. Algorithmic HRI has significantly improved the development of robots that can work for, around, and with humans. In this section, several important topics in HRI will be reviewed.

2.4.1 Prediction of Human Motion

Human motion prediction is a fundamental aspect of HRI. It has been a critical technique for many robotic applications, including self-driving cars, service robots, and unmanned aerial vehicles. Works on human state estimation and prediction generally include two categories: model-based and model-free approaches. For model-based approaches, filtering methods such as Kalman filters (KF) and particle filters (PF), have commonly been applied for predicting and tracking moving objects [94, 143, 180]. These methods first assume that people's motion can be described by certain dynamic model, such as the constant speed and direction model [154, 35], and then predict the human motion by using this model. However, when the human motion consists of multiple motion patterns, such as making turns, moving in curvature, and changing speed, such methods may fail to give an accurate prediction since they can only consider a single model of the person.

Learning techniques have also been utilized for human motion prediction in recent years. For example, Fulgenzi et al. [55] developed a Gaussian process-based motion predictor using pre-learned human motion patterns. Trautman et al. [161] proposed an interactive Gaussian Process approach for predicting the motion of a group of people. Human motion dynamics were modeled as a mixture of different kernels. Xiao et al. [177] predicted human future trajectory by using a set of homogeneous motion classes. The motion classes were obtained by using Support Vector Machine to classify a collection of observed human motion. Learning-based approaches can achieve good prediction performance if the environment where the person is walking is similar to the ones included in the training data, but may fail in unfamiliar environments. The lack of transferability has limited the use of these learning-based prediction methods.

2.4.2 Inference of Human Intention

Intent Recognition

In human-robot collaboration, the ability to infer an agent's intent composes a significant element of understanding its needs and generating collaborative actions accordingly. The Bayesian approach for intent recognition, which relates an agent's behavior with its underlying intent, has been adopted in plan recognition [29], human intention inference [9] and the perception of human action [184]. Some typical formalisms of intent recognition based on Bayesian approach include

Hidden Markov Models [58, 92], Dynamic Bayesian Networks (DBNs) [146] and Markov Decision Processes (MDPs) [49].

The key component in Bayesian approach is the relation describing an agent's action generation given its intent. The Boltzmann policy [9], or the Luce-Shepard choice rule [118], is one of the most basic models for such choice of behavior. It asserts that an agent is noisily optimal in behavior selection: the closer an action is to the optimal one in some metric (e.g. reward), the higher probability such action will be taken by the agent.

Intent Expression

Researchers have long studied how people use nonverbal communications in their interactions with one another [7, 63]. By making the robots' intended actions more readily apparent to their partners, we can improve people's abilities to coordinate their actions with that of robots. This line of research has led to works on anticipatory motion [61] and readable behavior [156]. Recent works have investigated different techniques for effectively expressing robots' intents, including motion trajectory planning [155], designing gesture [62], gaze [6, 126] and orientation [23]. Especially, people have incorporated animation principles [100, 157] to produce intentional motion, which focuses on pre- and post- action expressions of forethought and reaction to show robots' "thinking of action" [156], and attracting observer attention via exaggerated motion synthesis [61].

2.4.3 Observer-Aware Motion Planning

The motion planning problem focuses on generating a sequence of actions or trajectories that achieves some objectives while satisfying certain constraints. Recent works in human-robot interaction have started using motion as the communication channel for conveying a robot's plan, which gives rise to producing motion that is mindful of observer inferences. Such motion planners reason about the inferences that humans will make when observing the robot's behavior.

The understanding of a robot's intent is important for seamless human-robot collaboration. Researchers have studied speech for effective communication. However, language has its own ambiguities and can lead to miscommunication. Moreover, incessant explanations can be aggravating. Therefore, investigating the intent conveyance via actions is a natural and important topic for human-robot collaboration [160, 168, 136]. There exist two types of complementary inference that an observer can make to relate an agent's actions and goals: action-to-goal and goal-to-action [36]. The "action-to-goal" inference refers to the observer's ability to infer an agent's goal state based on the understanding of the function of an action; the "goal-to-action" inference, on the contrary, refers to an observer's ability to predict the actions that an agent will take given the knowledge of its goal. Traditional works that uses actions for intention inference have focused on inferring either human's next action or human's overall plans, and adapting the robot's plan in response [129, 116]. Effectively communicating a robot's next goal based on its ongoing motion has also been explored in robotics recently [46], which proposed algorithms for computing *legible* and *predictable* robot actions. Predictability usually refers to the property of motion that matches an observer's expectation when the goal of the motion is known to the observer. Some related works include [13, 4]. Legibility implies that an observer can understand or recognize the intent of the

robot by the knowledge of its action, when the observer does not know the robot's intent a priori. Some recent works on legible motion planning include [62, 42, 155].

2.4.4 Task Planning in Human-Robot Collaboration

Task planning in human-robot collaboration depends on the relation between the human and the robot during the interaction. Assistant robots are focused on the current task the human is conducting. When a need for assistance is detected, the robot plans for the best action to assist the human, given the current situation and human's intended task [163, 48]. Multi-modal communication is usually used for establishing and improving the understanding of human's needs [3].

For a human-robot team sharing a task, the planning usually involves each agent to decide and adapt their own collaborative actions based on other agents' behavior [28]. Tasks involving multiple subgoals, such as fetching objects from different positions and delivering them to specified location, usually requires effective assignment and scheduling of subgoals. Approaches have been developed for efficient online computation and adjustment of sequence of subgoals [149].

Robot's reasoning about human's intent also plays a significant role in task planning. Besides the intent recognition approaches described in Section 2.4.2, researchers have also investigated reasoning approached based on logic [174] and learned human models [128]. Such approaches have shown to be effective in enhancing task planning for human-robot collaboration.

Chapter 3

Distributed Bayesian Filter for Multi-Agent Systems

3.1 Introduction

Distributed filtering deals with using a group of networked robots to collectively estimate unknown environment states. The interaction topology plays a central role in the design of networked filters, of which two types are widely investigated in literature: centralized filters and distributed filters. In the former, local statistics estimated by each agent is transmitted to a single fusion center, where a global posterior distribution is calculated at each filtering cycle [186, 167]. In the latter, each agent individually executes distributed estimation, and the agreement of local estimates is achieved by certain consensus strategies [84, 140, 131]. In general, the distributed filters are more suitable in practice since they do not require a fusion center with powerful computation capability and are more robust to changes in network topology and link failures. So far, the distributed filters have two mainstream schemes in terms of the transmitted data among agents, i.e., *statistics dissemination* (SD) and *measurement dissemination* (MD). In the SD scheme, each agent exchanges statistics, such as posterior distributions and likelihood functions, within neighboring agents [76]. In the MD scheme, instead of exchanging statistics, each agent sends sensor measurements to neighboring agents.

The statistics dissemination scheme has been widely investigated during the last decade, especially in the field of signal processing, network control, and robotics. For example, Madhavan et al. presented a distributed extended Kalman filter for nonlinear systems [121]. This filter was used to generate local terrain maps by using pose estimates to combine elevation gradient and vision-based depth with environmental features. Olfati-Saber proposed a distributed linear Kalman filter (DKF) for estimating states of linear systems with Gaussian process and measurement noise [130]. Each DKF used additional low-pass and band-pass consensus filters to compute the average of weighted measurements and inverse-covariance matrices. Gu proposed a distributed particle filter for Markovian target tracking over an undirected sensor network [66]. Gaussian mixture models (GMM) were adopted to approximate the posterior distribution from weighted particles and the parameters of GMM were exchanged via average consensus filter. Hlinka et al. proposed a dis-

tributed method for computing an approximation of the joint (all-sensors) likelihood function by means of weighted-linear-average consensus algorithm when local likelihood functions belong to the exponential family of distributions [77]. Bandyopadhyay et al. presented a Bayesian consensus filter that uses logarithmic opinion pool for fusing posterior distributions of the tracked target [10]. Other examples can be found in [86] and [12].

Despite the popularity of statistics dissemination, exchanging statistics (e.g., probability density function, likelihood function) can cause high communication cost if the measured environment is relatively large in space and complicated in structure. This is because the probabilistic representation of states in such environments usually requires large support and cannot be readily represented by common families of distributions. Therefore it needs large amount of data to represent the distribution numerically. One remedy is to approximate the statistics with parametric models, e.g., Gaussian Mixture Model (GMM) [151], which can reduce communication burden to a certain extent as only parameters of GMM need to be transmitted. However, such manipulation increases the computation burden of each agent and sacrifices filtering accuracy due to approximation. The measurement dissemination scheme is an alternative solution to address the issue of exchanging large-scale statistics. An early work on measurement dissemination was done by Coates et al., who used adaptive encoding of observations to minimize communication overhead [34]. Ribeiro et al. exchanged quantized observations along with error-variance limits considering more pragmatic signal models [141]. A recent work was conducted by Djuric et al., who proposed to broadcast raw measurements to other agents, and therefore each agent has a complete set of observations of other agents for executing particle filtering [40]. A shortcoming of aforementioned works is that their communication topologies are assumed to be a complete graph: a robot can directly exchange information with any other robot via a single hopping even they are far from each other. This is a strong assumption on the communication network and is not always feasible in reality [84, 10]. For incomplete graphs, Leung et al. explored a decentralized filter for dynamic robot networks [107]. The algorithm was shown to achieve centralized-equivalent filtering performance in simulations. However, it required the communication of both measurements and statistics, which incurred large communication overhead.

The work in this chapter deals with these shortcomings. Specifically, the presented work develops distributed Bayesian filters (DBFs) using measurement dissemination scheme for a group of networked robots, while not requiring a complete graph. First, for time-invariant interaction (communication) networks, we propose a data exchange method called *Latest-In-and-Full-Out (LIFO)*, and develop a LIFO-based DBF (Section 3.3) for distributed filtering. Subsequently, we consider the distributed filter for time-variant networks in Section 3.4. We propose the *Full-In-and-Full-Out (FIFO)* method for data exchange and the *frequently jointly strongly connectedness* condition of the interaction network to guarantee the dissemination of sensor measurements in the network. We then develop a FIFO-based DBF for distributed filtering that achieves consistent estimation results.

3.2 Problem Formulation

Consider a network of N robots in a bounded two-dimensional space S, as shown in Fig. 3.1. Each robot is equipped with a sensor for target detection. Due to the limit of communication range, each

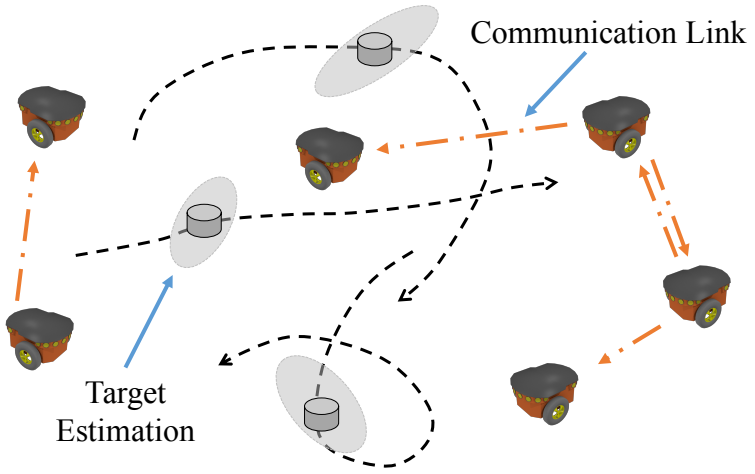


Figure 3.1: Target tracking scenario. The interaction topology is dynamically changing and each robot can only communicate with its neighboring robots.

robot can only exchange sensor measurements with its local neighbors. Every robot locally runs a Bayesian filter to estimate the target position in S utilizing its own measurements and the received measurements from other robots.

The target motion model uses a stochastic discrete-time model:

$$x^g_{k+1} = f^g(x^g_k) + v_k.$$

The sensor measurement is described by a stochastic model

$$z^i_k = h_i(x^g_k; x^i_k) + w^i_k, \tag{3.1}$$

where the superscript i is the index of the i^{th} robot.

3.2.1 Graphical Model of Interaction Topology

This work considers a simple[1] graph $G = (V, E)$ to represent the interaction topology of N networked robots, where the vertex set $V = \{1, \ldots, N\}$ represents the index set of robots and the edge set $E \subseteq V \times V$ denotes the communication links between robots. The *adjacency matrix* $A = \left[a_{ij}\right]$ of the graph G describes the interaction topology:

$$a_{ij} = \begin{cases} 1 & \text{if } (i, j) \in E \\ 0 & \text{if } (i, j) \notin E \end{cases},$$

where a_{ij} is the entry on the i^{th} row and j^{th} column of the adjacency matrix. The notation $a_{ij} = 1$ indicates that the i^{th} robot can directly communicate to the j^{th} robot and $a_{ij} = 0$ indicates no direct communication from i to j.

[1] A (directed/undirected) graph $G = (V, E)$ is *simple* if it has no self-loops (i.e., $(i, j) \in E$ only if $i \neq j$) or multiple edges with the same source and target nodes (i.e., E only contains distinct elements).

For an undirected graph, the adjacency matrix is symmetric. We define the *direct neighborhood* of i^{th} UGV as

$$\mathcal{N}_i = \left\{ j | a_{ij} = 1, \forall\, j \in V \right\}.$$

All the UGVs in $\mathcal{N}_i$ can directly exchange information with i^{th} UGV via single hopping. We also define another set $\mathcal{N}_i^a$, called *available neighborhood*, that contains indices of UGVs whose measurements can be received by the i^{th} UGV via single or multiple hoppings. Therefore $\mathcal{N}_i \subseteq \mathcal{N}_i^a$.

For a direct graph, we can similarly define the concept of neighborhood. But because the information flow is directional, there are two types of direct neighbors. Since in this dissertation work the direct graph is considered in the setting of time-variant networks, we use G_k to represent the interaction topology of time k. Let the *inbound neighbors* and *outbound neighbors* of the i^{th} robot under G_k be the set $\mathcal{N}_i^{\text{in}}(G_k) = \left\{ j | a_{ji}^k = 1,\ j \in V \right\}$ and $\mathcal{N}_i^{\text{out}}(G_k) = \left\{ j | a_{ij}^k = 1,\ j \in V \right\}$, respectively. All robots in $\mathcal{N}_i^{\text{out}}(G_k)$ can directly receive information from the i^{th} robot via the single hopping. A directed graph is *strongly connected* if there is a directed path connecting any two arbitrary vertices in V. Define the *union* of a set of simple directed graphs as the graph with the vertices in V and the edge set given by the union of each member's edge sets. Such collection is *jointly strongly connected* if the union of its members forms a strongly connected graph[2].

3.2.2 Communication Buffer

In this work we propose the concept of *communication buffer* (CB). Each robot maintains a CB that stores the state-measurement pairs of all robots. CBs are communicated among neighboring robots at each communication cycle. Let $Y_{\mathcal{K}}^i = \left\{ \left[x_k^i, z_k^i \right] | k \in \mathcal{K} \right\}$ be the set of state-measurement pairs of the i^{th} robot, where $\mathcal{K}$ is an index set of time steps. The i^{th} robot's *communication buffer* (CB) at time k is then defined as

$$\mathcal{B}_k^i = \left[Y_{\mathcal{K}_k^{i1}}^1, \ldots, Y_{\mathcal{K}_k^{iN}}^N \right], \tag{3.2}$$

where $\mathcal{K}_k^{ij} (j \in V)$ is the time index set indicating the times of j^{th} robot's state-measurement pairs that are contained in $\mathcal{B}_k^i$. Therefore, $Y_{\mathcal{K}_k^{ij}}^j$ represents the set of j^{th} robot's state-measurement pairs of time steps in $\mathcal{K}_k^{ij}$ that are stored in i^{th} robot's CB at time k. Note that the elements of $\mathcal{K}_k^{ij}$ differs between time-invariant and time-variant networks. The details of CB will be described in Sections 3.3 and 3.4.

3.2.3 Distributed Bayesian Filter

This part introduces the notations of DBF. Let $X_k \in S$ be the random variable representing the target position at time k. Define $\mathcal{Z}_k^i$ as the set of measurements that are made by some robots at time k and stored in the i^{th} robot's CB, i.e., $\mathcal{Z}_k^i = \left\{ z_k^j | \left[x_k^j, z_k^j \right] \in \mathcal{B}_k^i,\ \forall\, j \in V \right\}$. Let $\mathcal{Z}_{1:k}^i = \bigcup_{t=1}^{k} \mathcal{Z}_t^i$. It is easy to know that $\mathcal{Z}_{1:k}^i$ is the set of all measurements in $\mathcal{B}_k^i$. We also define $z_{1:k}^i = \left[z_1^i, \ldots, z_k^i \right]$ as the

[2]The counterpart definition for undirected graphs is given in [84].

set of the i^{th} robot's measurements at times 1 through k. The probability density function (PDF) of X_k, called *individual PDF*, of the i^{th} robot is represented by $P^i_{pdf}(X_k|\mathcal{Z}^i_{1:k})$. It is the estimation of the target position given all the measurements that the i^{th} robot has received. The initial individual PDF, $P^i_{pdf}(X_0)$, is constructed given prior information including past experience and environment knowledge. It is necessary to initialize $P^i_{pdf}(X_0)$ such that the probability density of the true target position is nonzero, i.e., $P^i_{pdf}(X_0 = x^g_0) > 0$. For example, a uniform distribution can satisfy.

Under the framework of DBF, the individual PDF is recursively estimated by two steps, as Section 2.2 describes. We rewrite them here using the notations specially for this chapter.

Prediction. At time k, the prior individual PDF $P^i_{pdf}(X_{k-1}|\mathcal{Z}^i_{1:k-1})$ is first predicted forward by using the Chapman-Kolmogorov equation:

$$P^i_{pdf}(X_k|\mathcal{Z}^i_{1:k-1}) = \int_{X_{k-1}\in S} P(X_k|X_{k-1})P^i_{pdf}(X_{k-1}|\mathcal{Z}^i_{1:k-1})dX_{k-1}, \tag{3.3}$$

where $P(X_k|X_{k-1})$ represents the state transition probability of the target.

Updating. The i^{th} individual PDF is then updated by the Bayes' rule using the set of measurements made by i^{th} robot and a subset of other robots at time k, i.e., $\mathcal{Z}^i_k$:

$$P^i_{pdf}(X_k|\mathcal{Z}^i_{1:k}) = K_i P^i_{pdf}(X_k|\mathcal{Z}^i_{1:k-1})P(\mathcal{Z}^i_k|X_k) \tag{3.4a}$$

$$= K_i P^i_{pdf}(X_k|\mathcal{Z}^i_{1:k-1}) \prod_{z^j_k\in\mathcal{Z}^i_k} P(z^j_k|X_k), \tag{3.4b}$$

where $P(z^j_k|X_k)$ is the sensor model and K_i is a normalization factor, given by:

$$K_i = \left[\int_{X_k\in S} P^i_{pdf}(X_k|\mathcal{Z}^i_{1:k-1})P(\mathcal{Z}^i_k|X_k)dX_k\right]^{-1}.$$

Here we have utilized the commonly adopted assumption [56, 66, 151] in the distributed filtering literature that the sensor measurement of each robot at current time is conditionally independent from its own previous measurements and the measurements of other robots given the target and the robot's current position. This assumption allows us to simplify $P(\mathcal{Z}^i_k|X_k, \mathcal{Z}^i_{1:k-1})$ as $P(\mathcal{Z}^i_k|X_k)$ in Eq. (3.4a) and factorize $P(\mathcal{Z}^i_k|X_k)$ into $\prod_{z^j_k\in\mathcal{Z}^i_k} P(z^j_k|X_k)$ in Eq. (3.4b).

3.3 Time-invariant Networks[3]

This section considers the distributed filtering under time-invariant interaction topologies with the application to target localization and tracking. The main contributions of the work in this section

[3]This section is based on the work published in [113].

include: (a) Different from existing works that assume full connectivity of the communication topology, each robot only needs to broadcast its CB to its neighbors via single hopping and then implements individual Bayesian filter locally using its own measurements and the ones transmitted from neighbors. (b) This work proposes the Latest-In-and-Full-Out (LIFO) protocol for information exchange among robots to reduce the communication burden, with the transmission data scaling linearly with the robot number. (c) The proposed LIFO-based DBF has the following properties: for a fixed and undirected network, LIFO guarantees the global dissemination of measurements over the network in a non-intermittent manner. The corresponding DBF ensures the consistency of estimated target position, i.e., the estimate converges in probability to the true value when the number of measurements tends to infinity.

3.3.1 Latest-In-and-Full-Out (LIFO) Method

The LIFO method is proposed for robots to exchange their CBs under undirected time-invariant networks. Under LIFO, each robot's CB stores its latest knowledge of state-measurement pairs of all robots. Using the notation in Eq. (3.2), this indicates $\mathcal{K}_k^{ij} = \left\{k_j^i\right\}$, where k_j^i is the latest time of j^{th} robot's sensor measurement that i^{th} robot has received by time k. Due to the communication delay, $k_j^i < k, \forall\, j \neq i$ and $k_i^i = k$ always hold.

The **LIFO** method is stated in Algorithm 2. As Corollary 1 will show, under LIFO, $\mathcal{N}_i^a = V \setminus \{i\}$[4], which indicates the desirable property that each robot can receive sensor measurements made by all other robots. Fig. 3.2 illustrates the LIFO cycles with 3 robots using a line topology. For general graphs, we have the following proposition:

Proposition 1 *For a fixed and undirected network of N robots, the latest measurements of i^{th} and j^{th} robot are exchanged via the shortest path(s) under LIFO. The delay of exchange $\tau_{i,j}$ is equivalent to the length of the shortest path(s) between them.*

Proof: Since the network is connected, there exists a minimum integer $\tau_{i,j}$ such that $a_{i,j}^{\tau_{i,j}} > 0$ and $\tau_{i,j}$ is the length of a shortest path between i^{th} and j^{th} robot [38]. Under LIFO, the latest measurement of i^{th} robot will always be propagated via the CBs of robots along a shortest path between i^{th} and j^{th} robot. Therefore, the latest measurement that j^{th} robot receives from i^{th} robot is delayed by $\tau_{i,j}$ rounds of communication. □

Corollary 1 *For the same topological assumption in Proposition 1, all elements in $\mathcal{B}_k^i$ under LIFO become nonempty when $k \geq N$. This implies $\mathcal{N}_i^a = V \setminus \{i\}$.*

Corollary 2 *For the same topology assumption in Proposition 1, once elements in $\mathcal{B}_k^i$ are nonempty, the updating of each element is non-intermittent. Moreover, $k - N < k_j^i \leq k,\ j \in V \setminus \{i\}$.*

Compared to statistics dissemination, LIFO is generally more communication-efficient for distributed filtering. To be specific, assume the environment S is represented by an $M \times M$ grid.

[4] '\' is the notation of set difference.

Algorithm 2 LIFO Method

(1) Initialization.
The CB of i^{th} robot is initialized as an empty set at $k = 0$:

$$\mathcal{B}_0^i = \left[Y^1_{\mathcal{K}_0^{i1}}, \ldots, Y^N_{\mathcal{K}_0^{iN}} \right], \text{ where } Y^j_{\mathcal{K}_0^{ij}} = \{[\varnothing, \varnothing]\}.$$

(2) At k^{th} step for i^{th} robot :
(2.1) Receiving Step.
The i^{th} robot receives all CBs from its direct neighborhood $\mathcal{N}_i$, each of which corresponds to the (k-1)-step CB of a robot in $\mathcal{N}_i$. The received CB from the l^{th} robot is $\mathcal{B}^l_{k-1}$ ($l \in \mathcal{N}_i^a$).

(2.2) Observation Step.
The i^{th} robot updates $Y^i_{\mathcal{K}_k^{ii}}$ by its own state-measurement pair:

$$Y^i_{\mathcal{K}_k^{ii}} = \left\{ \left[x_k^i, z_k^i \right] \right\}.$$

(2.3) Updating Step.
The i^{th} robot updates other entries of its own CB, $Y^j_{\mathcal{K}_k^{ij}}$ ($j \neq i$), by using all received CBs. Compare the element $(k-1)^i_j$ in $\mathcal{K}^{ij}_{k-1}$ with $(k-1)^l_j$ in $\mathcal{K}^{lj}_{k-1}$ for $\forall\, j \neq i,\ \forall l \in \mathcal{N}_i^a$ and find the most recent time index:

$$l_{\text{latest}} = \underset{\{l \in \mathcal{N}_i^a,\, i\}}{\operatorname{argmax}} \left\{ (k-1)^i_j, (k-1)^l_j \right\},$$
$$Y^j_{\mathcal{K}_k^{ij}} = Y^j_{\mathcal{K}^{l_{\text{latest}}j}_{k-1}},\ \mathcal{K}^{ij}_k = \mathcal{K}^{l_{\text{latest}}j}_{k-1},\ \forall\, j \neq i,\ \forall l \in \mathcal{N}_i^a.$$

(2.4) Sending Step.
The i^{th} robot broadcasts its updated CB to all of its neighbors defined in $\mathcal{N}_i$.
(3) $k \leftarrow k + 1$ until stop

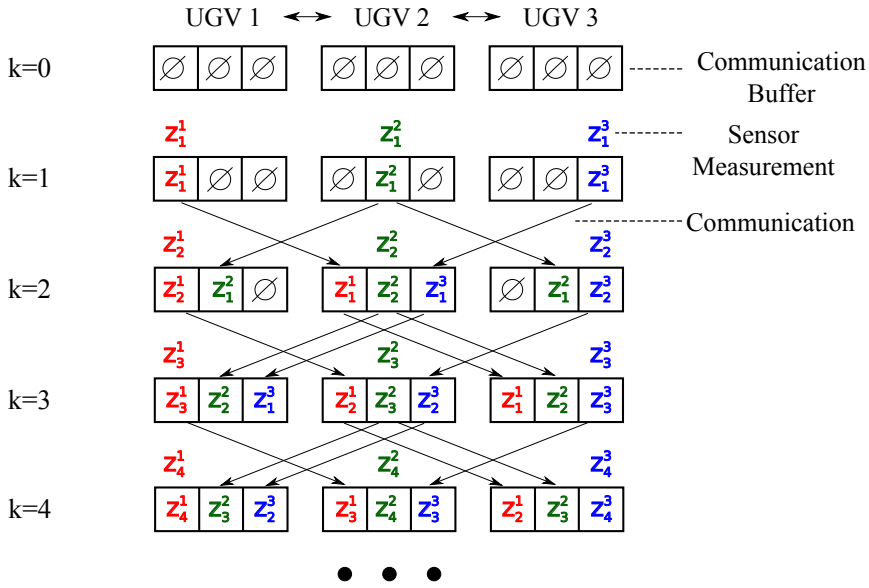


Figure 3.2: Example of LIFO with three robots using line communication topology. In this case, $\mathcal{N}_1 = \{2\}$, $\mathcal{N}_1^a = \{2,3\}$; $\mathcal{N}_2 = \{1,3\}$, $\mathcal{N}_2^a = \{1,3\}$; $\mathcal{N}_3 = \{2\}$, $\mathcal{N}_3^a = \{1,2\}$.

For a network of N robots, the transmitted data of LIFO between each pair of robots are only the CB of each robot and the corresponding sensor states, the size of which is $O(N)$, scaling linearly with robot number. On the contrary, the transmitted data for a statistics dissemination approach that transmits unparameterized posterior distributions or likelihood functions is $O(M^2)$, which is in the order of the environment size. Since M is generally much larger than N in applications such as target localization and environment monitoring, LIFO requires much less communication resources.

3.3.2 Algorithm of LIFO-DBF for Static Target

This section derives the LIFO-DBF algorithm for localizing a static target. It is assumed that all robots know the sensor model of other robots. The assumption of static target can simplify the Bayesian filter as the prediction step becomes trivial. At time k, the i^{th} robot starts from the last-step individual PDF that is locally stored, i.e., $P^i_{pdf}(X|\mathcal{Z}^i_{1:k-1})$[5]. The i^{th} individual PDF is then updated by fusing all newly received measurements, i.e., $\left\{z^1_{k^i_1}, \ldots, z^N_{k^i_N}\right\}$:

$$P^i_{pdf}(X|\mathcal{Z}^i_{1:k}) = K_i P^i_{pdf}(X|\mathcal{Z}^i_{1:k-1}) \prod_{j=1}^{N} P(z^j_{k^i_j}|X),$$

[5]Since the target is static, we drop the subscript k in X_k.

where

$$K_i = \left[\int_{X \in S} P^i_{pdf}(X|\mathcal{Z}^i_{1:k-1}) \prod_{j=1}^{N} P(z^j_{k^i_j}|X) dX \right]^{-1}.$$

The $P(z^j_{k^i_j}|X)$ is the measurement probability of the newly received sensor measurement that j^{th} robot made at time k^i_j.

3.3.3 Algorithm of LIFO-DBF for Moving Target

This section derives the LIFO-DBF for localizing a moving target. Instead of storing last-step PDF, at time k each robot maintains its individual PDF of time $(k-N)$, called the *stored PDF*. We use $P^i_{sto}(X_{k-N})$ to represent the i^{th} robot's stored PDF and $P^i_{sto}(X_{k-N}) = P^i_{pdf}(X_{k-N}|z^1_{1:k-N}, \ldots, z^N_{1:k-N})$ by definition. Each robot also needs to keep a collection of measurements that were made by robots from time $(k-N+1)$ to k and saved in the corresponding robot's CB. To be specific, let $\Omega^i_{\xi,k} = \left\{ j | j \in V, z^j_\xi \in \mathcal{Z}^i_{k-N+1:k} \right\}, \xi \in [k-N+1, k]$ be the index set of the measurements that are in the i^{th} CB in the time interval $[k-N+1, k]$. Then the collection of measurements that the i^{th} robot keeps is $\Omega^i_{k-N+1:k} = \cup^k_{t=k-N+1} \Omega^i_{t,k}$. The i^{th} individual PDF is then alternatively predicted and updated by using the aforementioned Bayesian filter (Eq. (3.3) and (3.4)) from $(k-N)$ to k to obtain $P^i_{pdf}(X_k|\mathcal{Z}^i_{1:k})$. Fig. 3.3 illustrates the LIFO-DBF procedure for the 1st robot as an example [6]. The **LIFO-DBF algorithm** for moving target is stated in Algorithm 3.

It is worth noting that, for the static target, each robot only needs current-step CB to update individual PDFs. Therefore, besides storing its own individual PDF of size $O(M^2)$, only current-step CB of size $O(N)$ is stored in an robot's memory and all previous CBs can be discarded, which means that the size of needed memory is $O(N + M^2)$. On the contrary, for the moving target, each robot needs to store a set of measurement history of size $O(N^2)$ and an individual PDF of size $O(M^2)$. Therefore the size of the needed memory for each robot is $O(M^2 + N^2)$. This is generally larger than that of statistics dissemination-based methods, the memory of which is $O(M^2)$. Besides, additional computation power is needed for LIFO-DBF compared to statistics dissemination-based methods. Therefore, LIFO-DBF sacrifices storage space and computation resource for reducing communication burden. This is actually desirable for real applications as local memory of vehicles is usually abundant compared to the limited bandwidth for communication.

3.3.4 Proof of Consistency

This section proves the consistency of the maximum a posteriori (MAP) estimator of LIFO-DBF under unbiased sensors (sensors without offset). An estimator of a state is said to be consistent if it converges in probability to the true value of the state [5]. Consistency is an important metric for stochastic filtering approaches [32] and it differs from the concept of consensus; consensus implies that the estimation results of all sensors converge to a same value, while consistency not only

[6] Due to the space limit, in this figure we use $P^i_{pdf}(k)$, $P^i_{pdf}(k-N)$ and $P^i_{pdf}(k-N+1)$ to represent $P^i_{pdf}(X_k|\mathcal{Z}^i_{1:k})$ $P^i_{pdf}(X_{k-N}|\mathcal{Z}^i_{1:k-N})$ and $P^i_{pdf}(X_{k-N+1}|\mathcal{Z}^i_{1:k-N+1})$, respectively.

Algorithm 3 LIFO-DBF Algorithm for Moving Target

For i^{th} robot at k^{th} step:
After updating CB by Algorithm 2,
(1) Initialize a *temporary PDF* by assigning the stored individual PDF to it:

$$P^i_{tmp}(X_{k-N}) = P^i_{sto}(X_{k-N}),$$

where the stored individual PDF is for time $(k-N)$:

$$P^i_{sto}(X_{k-N}) = P^i_{pdf}(X_{k-N}|z^1_{1:k-N}, \ldots, z^N_{1:k-N}).$$

(2) For $\xi = 1$ to N, iteratively repeat two steps of Bayesian filtering:
(2.1) Prediction

$$\begin{aligned} &P^{pre,i}_{tmp}(X_{k-N+\xi}) \\ &= \int_S P(X_{k-N+\xi}|X_{k-N+\xi-1}) P^i_{tmp}(X_{k-N+\xi-1}) dX_{k-N+\xi-1}. \end{aligned}$$

(2.2) Updating

$$P^i_{tmp}(X_{k-N+\xi}) = K_\xi P^{pre,i}_{tmp}(X_{k-N+\xi}) \prod_{j\in\Omega^i_{k-N+\xi,k}} P(z^j_{k-N+\xi}|X_{k-N+\xi}).$$

$$K_\xi = \left[\int_S P^{pre}_{tmp}(X_{k-N+\xi}) \prod_{j\in\Omega^i_{k-N+\xi,k}} P(z^j_{k-N+\xi}|X_{k-N+\xi}) dX_{k-N+\xi} \right]^{-1}.$$

(2.3) When $\xi = 1$, if $z^j_{k-N+1} \neq \emptyset$ for $\forall j \in \mathcal{N}^a_i$, update the stored PDF:

$$P^i_{sto}(X_{k-N+1}) = P^i_{tmp}(X_{k-N+1}),$$

where

$$P^i_{tmp}(X_{k-N+1}) = P^i_{pdf}(X_{k-N+1}|z^1_{1:k-N+1}, \ldots, z^N_{1:k-N+1}).$$

(3) Individual PDF of i^{th} robot at time k is $P^i_{pdf}(X_k|\mathcal{Z}^i_{1:k}) = P^i_{tmp}(X_k)$.

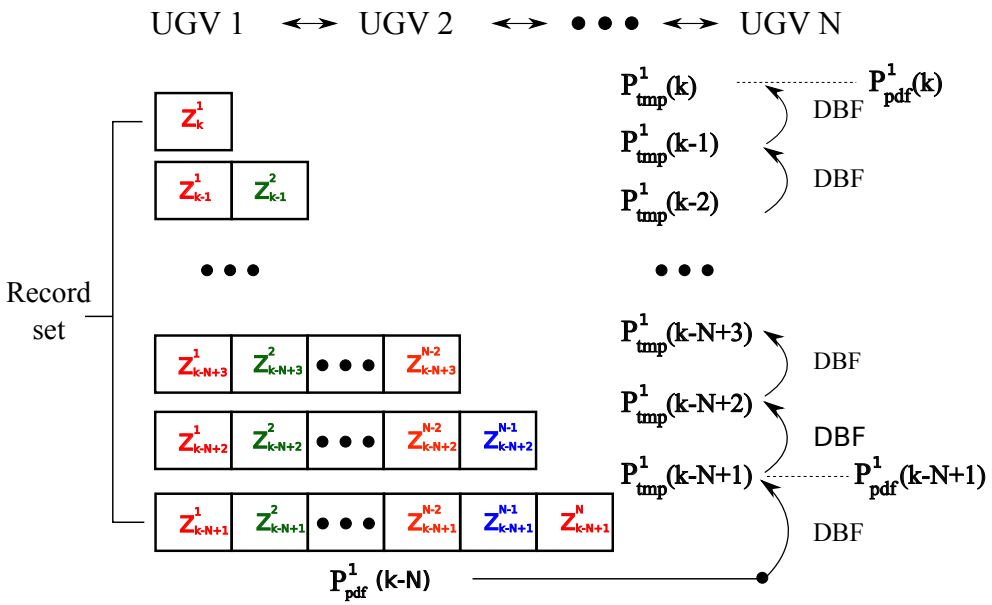


Figure 3.3: Example of LIFO-DBF for 1st robot at time k. Networked robots take a line topology. The stored individual PDF is represented by $P^1_{pdf}(k-N)$. The robot first calculates $P^1_{tmp}(k-N+1)$, defined in Algorithm 3, and then stores it as $P^1_{pdf}(k-N+1)$. Repeat DBF until $P^1_{pdf}(k)$ is obtained. In this example, $\Omega^1_{k-N+\xi} = \{1, 2, \ldots, N+1-\xi\}$, $\xi = 1, \ldots, N$.

implies achieving consensus asymptotically, but also requires that the estimated value converge to the true value. Here we prove the consistency for static robots.

The consistency of LIFO-DBF for static robots is stated as follows:

Theorem 1 *Assume the robots are static and the sensors are unbiased, if the network of N robots is connected and time-invariant, then the MAP estimator of target position converges in probability to the true position of the target using LIFO-DBF, i.e.,*

$$\lim_{k\to\infty} P(X_k^{MAP,i} = x^g) = 1, \ i \in V,$$

where

$$X_k^{MAP,i} = \arg\max_X P^i_{pdf}(X|\mathcal{Z}^i_{1:k}).$$

Proof: The DBF can be transformed into the batch form by recursively applying Eq. (3.4) from k to the initial time 1 (back in time):

$$\begin{aligned}
P^i_{pdf}(X|\mathcal{Z}^i_{1:k}) &= K_i P^i_{pdf}(X|\mathcal{Z}^i_{1:k-1}) \prod_{z^j_k \in \Omega^i_{k,k}} P(z^j_k|X) \\
&= K_i P^i_{pdf}(X|\mathcal{Z}^i_{1:k-2}) \prod_{z^j_{k-1} \in \Omega^i_{k-1,k}} P(z^j_{k-1}|X) \prod_{z^j_k \in \Omega^i_{k,k}} P(z^j_k|X) \\
&= \dots \\
&= K_i P^i_{pdf}(X) \prod_{z^j_1 \in \Omega^i_{1,k}} P(z^j_1|X) \cdots \prod_{z^j_k \in \Omega^i_{k,k}} P(z^j_k|X) \\
&= K_i P^i_{pdf}(X) \prod_{j=1}^{N} \prod_{t \in \mathcal{K}^{ij}_k} P(z^j_t|X).
\end{aligned}$$

The last step is obtained by using the relation $\mathcal{B}^i_k = \left[Y^1_{\mathcal{K}^{i1}_k}, \dots, Y^N_{\mathcal{K}^{iN}_k} \right]$ and $\mathcal{Z}^i_{1:k}$ is the set of all measurements in $\mathcal{B}^i_k$.

Comparing $P^i_{pdf}(X_k = x|\mathcal{Z}^i_{1:k})$ with $P^i_{pdf}(X_k = x^g|\mathcal{Z}^i_{1:k})$[7] yields

$$\frac{P^i_{pdf}(x|\mathcal{Z}^i_{1:k})}{P^i_{pdf}(x^g|\mathcal{Z}^i_{1:k})} = \frac{P^i_{pdf}(x) \prod_{j=1}^{N} \prod_{t \in \mathcal{K}^{ij}_k} P(z^j_t|x)}{P^i_{pdf}(x^g) \prod_{j=1}^{N} \prod_{t \in \mathcal{K}^{ij}_k} P(z^j_t|x^g)}. \tag{3.5}$$

Take the logarithm of Eq. (3.5) and average it over k steps:

$$\frac{1}{k} \ln \frac{P^i_{pdf}(x|\mathcal{Z}^i_{1:k})}{P^i_{pdf}(x^g|\mathcal{Z}^i_{1:k})} = \frac{1}{k} \ln \frac{P^i_{pdf}(x)}{P^i_{pdf}(x^g)} + \sum_{j=1}^{N} \frac{1}{k} \sum_{t \in \mathcal{K}^{ij}_k} \ln \frac{P(z^j_t|x)}{P(z^j_t|x^g)}. \tag{3.6}$$

Since $P^i_{pdf}(x)$ and $P^i_{pdf}(x^g)$ are bounded and nonzero by the choice of the initial PDF,

$$\lim_{k \to \infty} \frac{1}{k} \ln \frac{P^i_{pdf}(x)}{P^i_{pdf}(x^g)} = 0.$$

Note that the sensor measurement is a random variable drawn from the underlying distribution associated with the sensor model, i.e., $z^j_t \sim P(\cdot|x^g),\ j \in V$. Therefore $\ln \frac{P(z^j_t|x)}{P(z^j_t|x^g)}$ is a random variable associated with z^j_t. Due to the finite delay of measurement arrival (Corollary 2), i.e., $k - N < k^i_j \le k$,

[7] For the purpose of simplicity, we use $P^i_{pdf}(x|\mathcal{Z}^i_{1:k})$ to represent $P^i_{pdf}(X_k = x|\mathcal{Z}^i_{1:k})$ in this proof.

where $|\cdot|$ is the set cardinality, we can use the law of large numbers to study the asymptotic behavior of the series in Eq. (3.6):

$$\frac{1}{k}\sum_{t\in\mathcal{K}_k^{ij}} \ln\frac{P(z_t^j|x)}{P(z_t^j|x^g)} \xrightarrow{P} \mathbb{E}_{z_t^j}\ln\left[\frac{P(z_t^j|x)}{P(z_t^j|x^g)}\right] \tag{3.7a}$$

$$= \int_{z_t^j} P(z_t^j|x^g)\ln\frac{P(z_t^j|x)}{P(z_t^j|x^g)}dz_t^j \tag{3.7b}$$

$$= -D_{KL}\left(P(z_t^j|x)\|P(z_t^j|x^g)\right), \tag{3.7c}$$

where "$\xrightarrow{P}$" represents "convergence in probability" and $D_{KL}(P_1\|P_2)$ denotes the Kullback-Leibler (KL) divergence between two probability distribution P_1 and P_2. KL divergence has the property that $\forall P_1,\ P_2,\ D_{KL}(P_1\|P_2) \le 0$, and the equality holds if and only if $P_1 = P_2$. Therefore

$$\lim_{k\to\infty}\frac{1}{k}\sum_{t\in\mathcal{K}_k^{ij}}\ln\frac{P(z_t^j|x)}{P(z_t^j|x^g)} < 0, \quad x \neq x^g$$

$$\lim_{k\to\infty}\frac{1}{k}\sum_{t\in\mathcal{K}_k^{ij}}\ln\frac{P(z_t^j|x)}{P(z_t^j|x^g)} = 0, \quad x = x^g.$$

Considering the limiting case of Eq. (3.6), we get

$$\lim_{k\to\infty}\frac{1}{k}\ln\frac{P^i_{pdf}(x|\mathcal{Z}^i_{1:k})}{P^i_{pdf}(x^g|\mathcal{Z}^i_{1:k})} < 0, \quad x \neq x^g \tag{3.8}$$

$$\lim_{k\to\infty}\frac{1}{k}\ln\frac{P^i_{pdf}(x|\mathcal{Z}^i_{1:k})}{P^i_{pdf}(x^g|\mathcal{Z}^i_{1:k})} = 0, \quad x = x^g. \tag{3.9}$$

Eq. (3.8) and (3.9) imply that

$$\frac{P^i_{pdf}(x|\mathcal{Z}^i_{1:k})}{P^i_{pdf}(x^g|\mathcal{Z}^i_{1:k})} \xrightarrow{P} \begin{cases} 0 & x \neq x^g, \\ 1 & x = x^g. \end{cases}$$

Therefore,

$$\lim_{k\to\infty} P(X_k^{MAP} = x^g) = 1.$$

□

Remark 1 *The assumption of unbiased sensors are important for the consistency of the estimator. In fact, with unknown non-zero bias, the distribution of z_l^j differs from $P(z_l^j|x^g)$, which invalidates the derivation in Eq.* (3.7) *and the consistency proof. This assumption also makes intuitive sense. In the extreme case, if each sensor has a very large unknown measurement offset, then the estimated target position of each sensor (without communicating with other sensors) will be very different from each other's. Therefore, no common target position can be correctly obtained when they fuse measurements.*

3.3.5 Simulation and Experiment

Simulation has been conducted to evaluate the effectiveness of LIFO-DBF for target localization using a team of six robots. The networked robots take a ring communication topology that each robot can communicate with two fixed neighbors. DBF is implemented using the histogram filter method, by which the continuous space is finely discretized into finitely many regions and the individual PDF is approximated by the cumulative probability of each region [158]. Two types of scenarios are used: in the first scenario, both robots and the target are static; the second scenario subsequently deals with moving robots for localizing a moving target. In the first scenario, we test two different kinds of robot team: the first robot team, called the homogeneous team, is equipped with bearing-only sensors; in the second team, called the heterogeneous team, three robots are equipped with bearing-only sensor and the other three equipped with range-only sensors. In the second scenario, the heterogeneous team is tested. The bearing-only sensors are assumed to have large enough sensing range to cover the simulation field and the measurement noise is a zero-mean Gaussian white noise with standard deviation being 0.5. The range-only sensors are assumed to have 360° field of view and the measurement noise is a zero-mean Gaussian white noise with standard deviation being 5.

In both scenarios, LIFO-DBF is compared with two commonly adopted approaches in multi-agent filtering: the consensus-based distributed filtering (CbDF) method [132] and the centralized filtering (CF) method [166]. The CbDF requires robots to continually exchange their individual PDFs with direct neighbors, computing the average of its own and the received PDFs. Multiple rounds of communication and averaging are conducted at each step to ensure the convergence of robot's individual PDFs. The CF assumes a central unit that can constantly receive and fuse all robots' latest measurements into a single PDF. Ten test trials with randomly generated initial robot and target positions are run and each trial is terminated after 50 time steps. We quantitatively compare the three filters in terms of the estimation error and entropy of the uncertainty.

The estimation error is defined as the difference between the true target position and the estimated position (using the MAP estimate of the individual PDF):

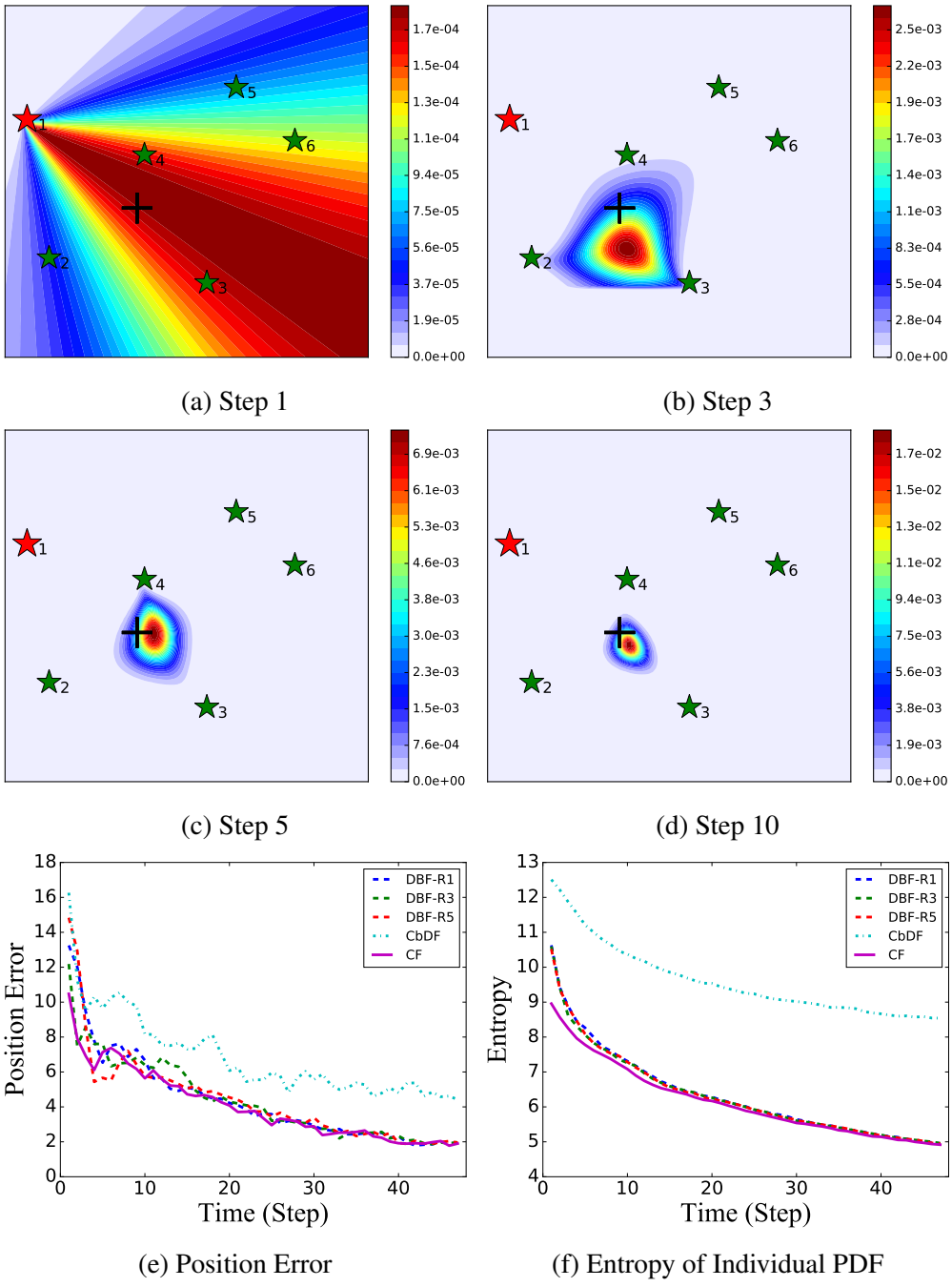
$$\Delta_k = \|X_k^{\mathrm{MAP}} - x_k^g\|_2.$$

The entropy of the uncertainty is

$$H_k = \sum_{X_k \in S} -P_{pdf}(X_k) \log(P_{pdf}(X_k)).$$

Static robots, Static Target

The individual PDF of each robot is initialized as a uniform distribution over a bounded space. At each time step, each robot executes the LIFO-DBF for static target (Section 3.3.2) to update their estimation of target position. The evolution of the 1st robot's individual PDF is shown in Figures 3.4a to 3.4d. Since the robot is equipped with a noisy bearing-only sensor, the measurement at step 1 results in a wide distribution with the peak centered along the noise-corrupted measured direction from the sensor to the target, as Figure 3.4a illustrates. As more measurements are fused,

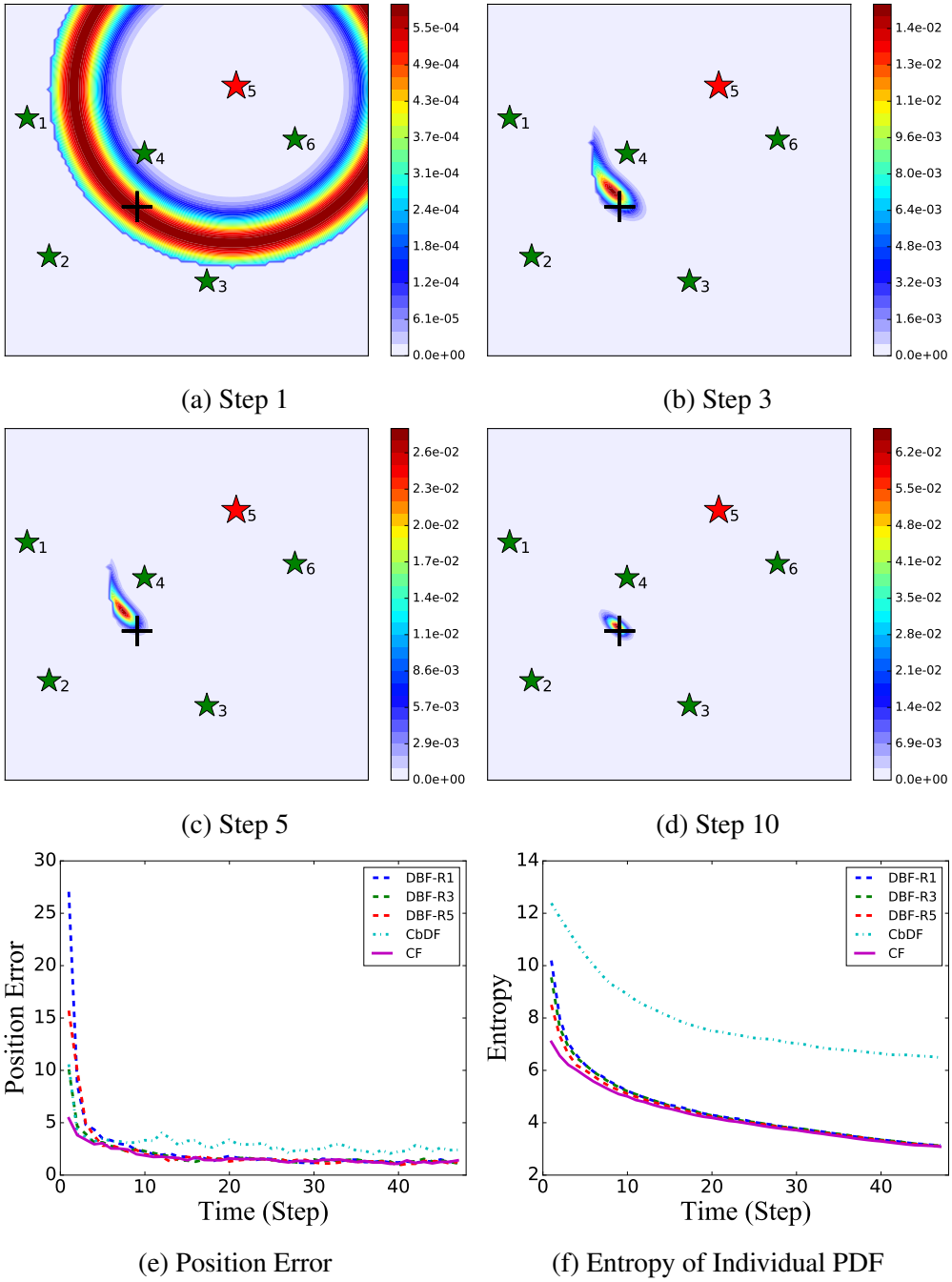


(a) Step 1 (b) Step 3

(c) Step 5 (d) Step 10

(e) Position Error (f) Entropy of Individual PDF

Figure 3.4: Simulation results for LIFO-DBF with static target and static robots. (a)-(d) The 1st robot's individual PDF at different times. The black cross stands for the target. Note that the value of the color bar varies among different figures. (e) Average position estimation error of the 1st, 3rd and 5th robot's LIFO-DBF, the CbDF and the CF. (f) Average entropy of individual PDFs.

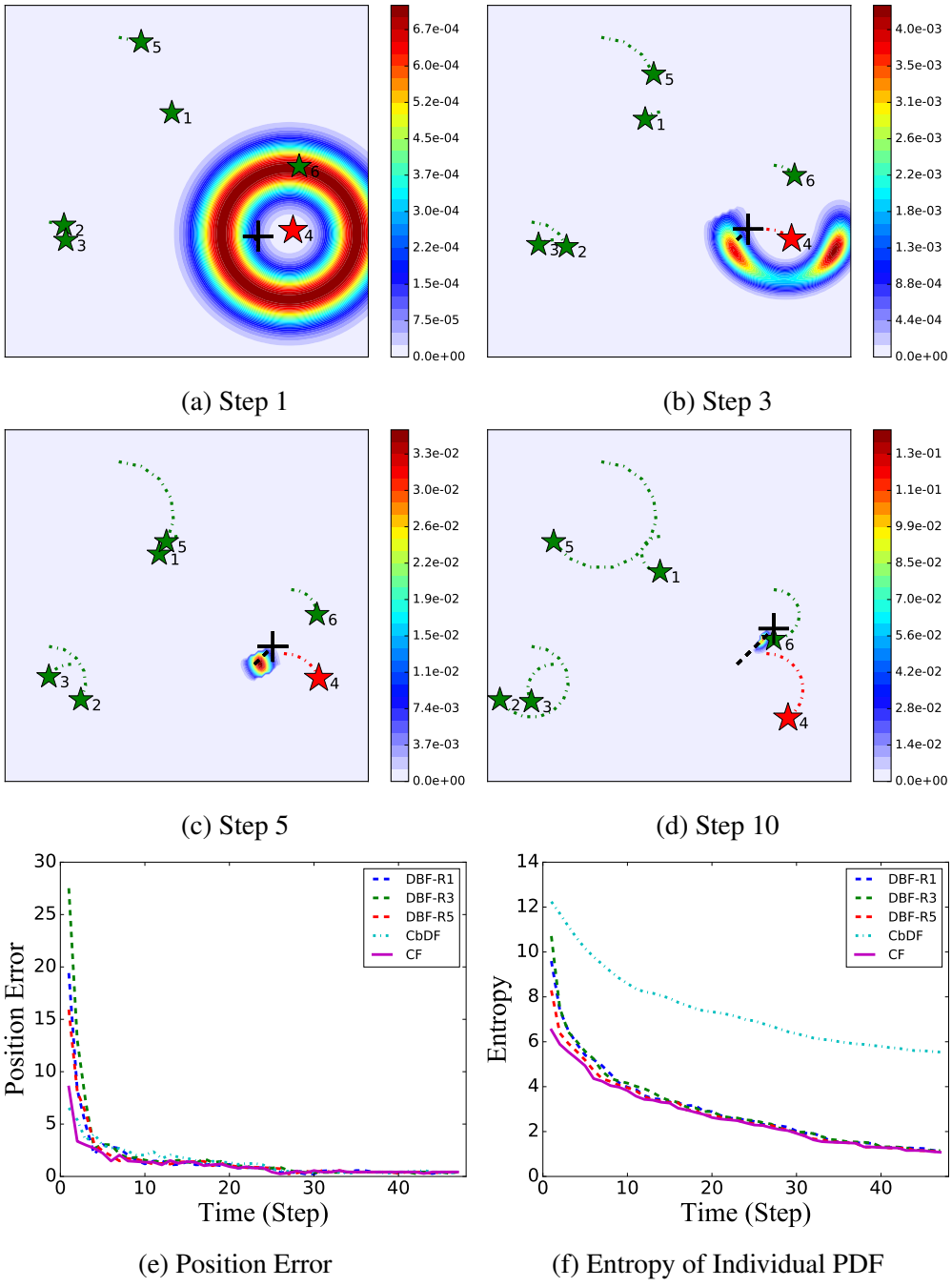


(a) Step 1 (b) Step 3

(c) Step 5 (d) Step 10

(e) Position Error (f) Entropy of Individual PDF

Figure 3.5: (a)-(d) The 5th robot's individual PDF at different times. (e) Average position estimation error. (f) Average entropy of individual PDFs.



(e) Position Error (f) Entropy of Individual PDF

Figure 3.6: (a)-(d) The 4th robot's individual PDF at different times. (e) Average position estimation error. (f) Average entropy of individual PDFs.

the individual PDF asymptotically concentrates to the true location of the target, which accords with the consistency of LIFO-DBF.

The Figures 3.4e and 3.4f shows the comparison of LIFO-DBF with CbDF and CF. Unsurprisingly, the CF achieves the best performance in terms of both small position estimation error and fast reduction of entropy. This happens because the central unit has access to the latest measurements of all robots, thus being able to make the most use of all available information. It is worth noting that, LIFO-DBF achieves similar asymptotic performance as the CF, both in position estimation error and entropy reduction; this is achieved even though each robot only communicates with its two neighboring robots, which requires less communication burden than the CF. The CbDF has the worst performance among these three filtering approaches. It results in slow reduction of entropy and the position error remains large. This happens because Bayes filtering (Eq. (3.3) and (3.4)) is a nonlinear process. Using the linear average consensus law to fuse individual PDFs thus deviates from the actual Bayesian filtering and therefore cannot fully exploit the information of new measurements to reduce the uncertainty of estimation.

The Figure 3.5 shows the simulation results of the heterogeneous team. The individual PDF of the robot with a noisy range-only sensor is presented in Figures 3.5a to 3.5d. At step 1, the target is detected and the individual PDF is updated such that the peak of the distribution centers at the positions whose distance to the sensor equals the noise-corrupted measured value. Similar to the case of using the homogeneous team, the individual PDF asymptotically concentrates to the true target position. Due to the use of various sensors, the estimation accuracy increases faster than the homogeneous team case, as shown in Figures 3.5e and 3.5f. It can also be noticed that, LIFO-DBF achieves similar performance as CF, while the performance of CbDF is the worst.

Moving robots, Moving Target

In this scenario, each robot follows a pre-defined circular trajectory. The target motion is modeled as a single-integrator with unit velocity on each coordinate. Each robot executes the LIFO-DBF for moving target (Section 3.3.3) for estimation.

Figs. 3.6a to 3.6d illustrate the evolution of individual PDF for the heterogeneous team. They present similar asymptotic behavior of individual PDFs as in previous simulations. Figs. 3.6e and 3.6f compare LIFO-DBF with CbDF and CF. It is worth noting that CbDF achieves comparable position estimation error performance as the CF and LIFO-DBF. However, CbDF requires multiple rounds of exchanging individual PDFs, which incurs much higher communication burden than LIFO-DBF at each time step. Considering the small difference in position estimation error and significantly faster entropy reduction, LIFO-DBF is still preferable over CbDF for moving target scenario.

Indoor Experiment using Mobile Robot

A target localization experiment has been conducted to validate the LIFO-DBF approach in an indoor environment. The experiment equipments are illustrated in Fig. 3.7. Each robot is equipped with an onboard sonar sensor to measure the target (a small cardboard box) position. The sonar sensor has the sensing field of view of 25° and the range of $4.9m$. Its measurement noise is modeled

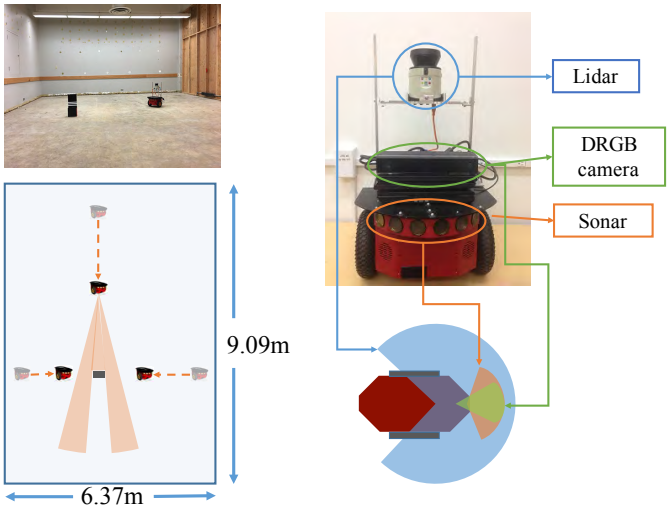


Figure 3.7: Experiment setup. Left: Three robots are used for localizing the target (black box). The room size is $9.09m \times 6.37m$. The orange sectors in the bottom figure show the sonar sensing domain. Red dashed lines show the trajectory of each robot. Right: The mobile robot platform is equipped with different sensors. The sonar sensor is used for target localization.

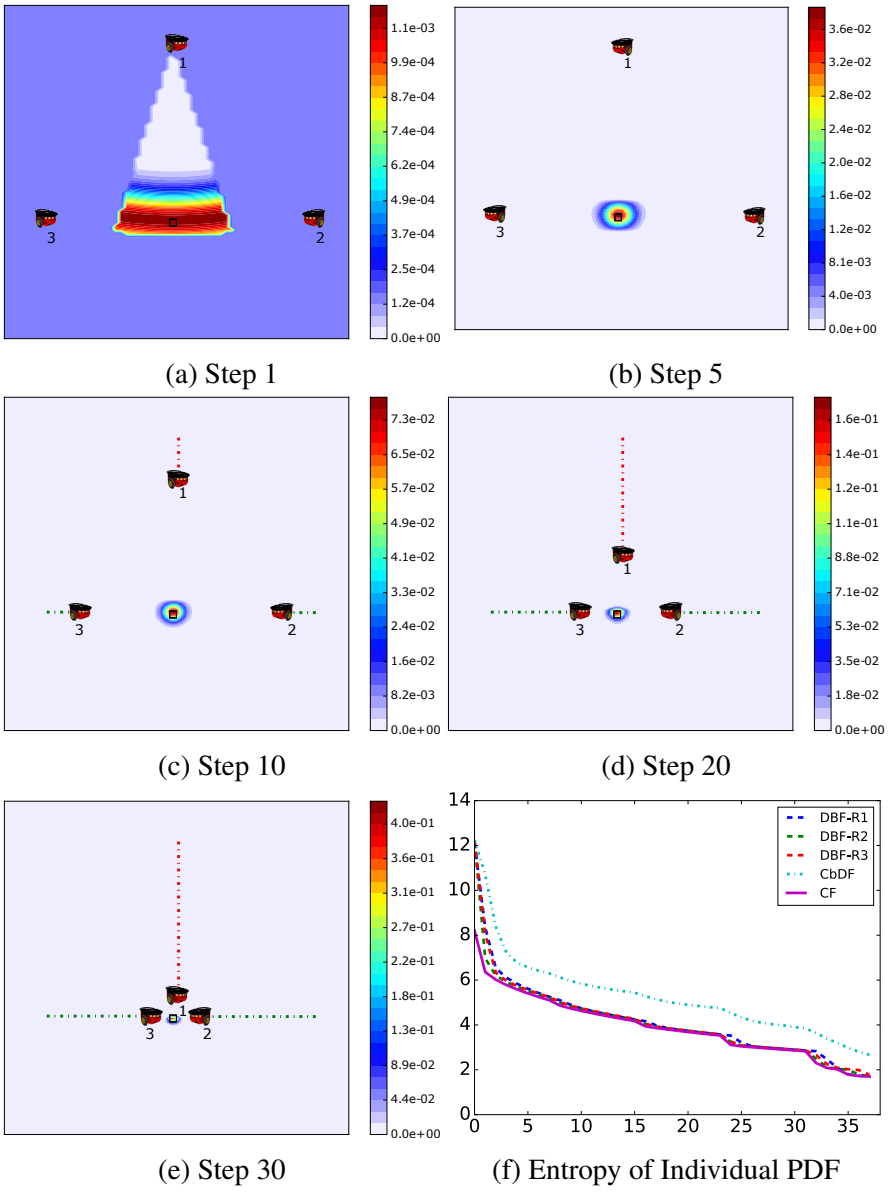
as a zero-mean Gaussian random variable with standard deviation $0.51m$. The robots locally run the LIFO-DBF to estimate the target position. Each robot's individual PDF is constructed on an evenly spaced grid with $0.1m$ interval on each axis. In the experiment, robots move to different locations to measure the target position. We manually measure the robot positions where the sensor measurements are obtained to reduce the effects of localization error. The effect of the target size has been compensated during the post-processing of the measurement data.

Figs. 3.8a to 3.8e show the evolution of the 1st robot's individual PDF. At step 1, the probability mass concentrates to a stripe due to the fact that the sonar sensor used in this experiment has limited FOV and noise in range measurement. As new measurements from different robots are continually fused, the probability mass gradually concentrates to the true position of the target. Similar to previous simulations, we compare our method with CbDF and CF. All three approaches achieve accurate position estimation. However, they differ in the uncertainty reduction, as shown in Fig. 3.8f. The CF has the fastest entropy reduction and the LIFO-DBF achieves comparable performance, while the CbDF shows the slowest entropy reduction. These experiment results validate the consistency of LIFO-DBF and demonstrate its effectiveness for distributed filtering.

3.4 Time-variant Networks[8]

In many real applications, the interaction topology can be time-varying due to unreliable links, external disturbances or range limits [176]. In such cases, dynamically changing topologies can cause random packet loss, variable transmission delay, and out-of-sequence measurement (OOSM)

[8]This section is based on the work published in [115].



(a) Step 1 (b) Step 5

(c) Step 10 (d) Step 20

(e) Step 30 (f) Entropy of Individual PDF

Figure 3.8: The 1st robot's individual PDF at different time steps.

issues [175], thus negatively affecting the performance of distributed estimation. This section proposes a new distributed Bayesian filtering (DBF) method for a group of networked robots with dynamically changing interaction topologies. For the purpose of narrative simplicity, the directed graph is used in this work. The undirected graphs can actually be treated as bidirectional directed graphs. The work presented in this section makes the following contributions: (1) We introduce a new protocol called the Full-In-and-Full-Out (FIFO) that allows each robot to broadcast a history of measurements to its neighbors via single hopping, enabling the localization and tracking of targets using general nonlinear sensor models under time-varying topologies. (2) We propose the *frequently jointly strongly connectedness* condition of the interaction topology and show that, under this condition, FIFO can disseminate robots' measurements over the network within a finite time. (3) We develop a FIFO-based distributed Bayesian filter (FIFO-DBF) for each robot to implement locally. A track list is designed to reduce the computational complexity of FIFO-DBF and the communication burden. The FIFO-DBF can avoid the OOSM issue. (4) We prove the consistency of FIFO-DBF for localizing a static target: each robot's estimate of target position converges in probability to the true target position asymptotically if the interaction topologies are *frequently jointly strongly connected.*

3.4.1 Full-In-and-Full-Out (FIFO) Method

We proposes a Full-In-and-Full-Out (FIFO) method for measurement exchange in dynamically changing interaction topologies. The use of FIFO allows us to apply measurement-dissemination-based distributed filters to time-varying networks. Each robot contains a communication buffer (CB) and a track list (TL). The CB is defined in Eq. (3.2). A robot's TL stores the information of this robot's reception of all robots' measurements, and is used for trimming old state-measurement pairs in the CB to reduce the communication burden. The details of TL will be introduced in Section 3.4.3. Each robot sends its CB and TL to its outbound neighbors at every time step.

The **FIFO protocol** is stated in Algorithm 4, which consists of the CB and TL parts. For the purpose of clarity, we focus on the CB parts in this section and leave the description of the TL parts to Section 3.4.3. Figure 3.9 illustrates the FIFO cycles in a network of three robots with dynamically changing topologies. The following facts can be observed from Figure 3.9: (1) the topologies are jointly strongly connected in the time interval $[0, 6)$; (2) each robot can receive the state-measurement pairs of other robots within finite steps. We extend these facts to a network of N robots:

Definition 1 (frequently jointly strongly connected topologies) *Consider a network of N robots with dynamically changing interaction topologies* $\mathbb{G} = \{G_1, G_2, G_3 \ldots, \}$. *If* $\mathbb{G}$ *satisfies the following conditions:*

1. *there exists an infinite sequence of time intervals* $[k_m, k_{m+1})$, $m = 1, 2, \ldots,$ *starting at* $k_1 = 0$ *and are contiguous, nonempty, and uniformly bounded;*

2. *the union of graphs across each such interval is jointly strongly connected,*

then $\mathbb{G}$ *is a set of frequently jointly strongly connected topologies.*

Algorithm 4 FIFO Protocol

(1) Initialization.
CB: The CB of i^{th} robot is initialized as an empty set at $k = 0$:

$$\mathcal{B}_0^i = \left[Y^1_{\mathcal{K}_0^{i1}}, \ldots, Y^N_{\mathcal{K}_0^{iN}} \right], \text{ where } Y^j_{\mathcal{K}_0^{ij}} = \{[\varnothing, \varnothing]\}.$$

TL: The TL of i^{th} robot is initialized at $k = 0$:

$$\mathbf{q}_1^{ij} = [0, \ldots, 0, 1], \text{ i.e. } q_1^{jl} = 0, \forall j, l \in \{1 \ldots, N\}.$$

(2) At time k ($k \geq 1$) for i^{th} robot:
(2.1) Receiving Step.
CB: The i^{th} robot receives all CBs of its inbound neighbors $\mathcal{N}_i^{\text{in}}(G_{k-1})$. The received CB from the l^{th} robot is $\mathcal{B}_{k-1}^l$ ($l \in \mathcal{N}_i^{\text{in}}(G_{k-1})$).
TL: The i^{th} robot receives all TLs of its inbound neighbors $\mathcal{N}_i^{\text{in}}(G_{k-1})$. The received TL from the l^{th} robot is Q_{k-1}^l ($l \in \mathcal{N}_i^{\text{in}}(G_{k-1})$).

(2.2) Observation Step.
CB: The i^{th} robot updates $Y^i_{\mathcal{K}_k^{ii}}$ by its own state-measurement pair:

$$Y^i_{\mathcal{K}_k^{ii}} = Y^i_{\mathcal{K}_{k-1}^{ii}} \cup \left\{ \left[x_k^i, z_k^i \right] \right\}.$$

(2.3) Updating Step.
CB: The i^{th} robot updates other entries of its own CB, $Y^j_{\mathcal{K}_k^{ij}}$ ($j \neq i$), by merging with all received CBs:

$$Y^j_{\mathcal{K}_k^{ij}} = Y^j_{\mathcal{K}_{k-1}^{ij}} \cup Y^j_{\mathcal{K}_{k-1}^{lj}}, \ \forall j \neq i, \ \forall l \in \mathcal{N}_i^{\text{in}}(G_{k-1}).$$

TL: The i^{th} robot updates its own TL, Q_k^i, using the received TLs (see Algorithm 6). The CB is trimmed based on the updated track list (see Algorithm 7).
(2.4) Sending Step.
CB: The i^{th} robot sends its updated CB, $\mathcal{B}_k^i$, to all of its outbound neighbors defined in $\mathcal{N}_i^{\text{out}}(G_k)$.
TL: The i^{th} robot sends its updated track list, Q_k^i, to its outbound neighbors $\mathcal{N}_i^{\text{out}}(G_k)$.
(3) $k \leftarrow k + 1$ until stop

Theorem 2 *Consider a network of N robots with dynamically changing interaction topologies $\mathbb{G}$. If $\mathbb{G}$ is frequently jointly strongly connected, then each pair of robots can exchange measurements under FIFO. In addition, it takes no more than NT_u steps for a robot to communicate to another one, where $T_u = \sup_{m=1,2,\ldots} (k_{m+1} - k_m)\, T$ is the upper bound of interval lengths.*

Proof: Without loss of generality, we consider the transmission of $\mathcal{B}^i_{t_1}$ from the i^{th} robot to an arbitrary j^{th} robot ($j \in V \backslash \{i\}$), where $t_1 \in [k_1, k_2)$. Since each robot will receive inbound neighbors' CBs and send the merged CB to its outbound neighbors at the next time step, the i^{th} robot can transmit $\mathcal{B}^i_{t_1}$ to j^{th} robot if and only if a path $[l_1, \ldots, l_n]$ exists, with $l_1 = i$, $l_n = j$, $l_2, \ldots, l_{n-1} \in V \setminus \{i, j\}$, and the edges (l_s, l_{s+1}) appears no later than (l_{s+1}, l_{s+2}), $s = 1, \ldots, n-2$.

As the union of graphs across the time interval $[k_2, k_3)$ is jointly connected, i^{th} robot can directly send $\mathcal{B}^i_{t_1}$ to at least one another robot at a time instance, i.e., $\exists l_2 \in V \setminus \{i\}$, $\exists t_2 \in [k_2, k_3)$ s.t. $l_2 \in \mathcal{N}_i^{\text{out}}(G_{t_2})$. If $l_2 = j$, then $\mathcal{B}^i_{t_1}$ has been sent to j. If $l_2 \neq j$, $\mathcal{B}^i_{t_1}$ has been merged into $\mathcal{B}^{l_2}_{t_2+1}$ and will be sent out in the next time step.

Using the similar reasoning for time intervals $[k_m, k_{m+1})$, , $m = 3, 4, \ldots$, it can be shown that all robots can receive the state-measurement pairs in $\mathcal{B}^i_{t_1}$ no later by k_{N+1}. Therefore, the transmission time from an arbitrary robot to any other robots is no greater than NT_u.

□

Corollary 3 *For a frequently jointly strongly connected network, each robot receives the CBs of all other robots under FIFO within finite time.*

Proof: According to Theorem 2, each robot is guaranteed to receive $\mathcal{B}^j_t$ ($\forall t \geq 0$, $j \in V$) when $k \geq t + NT_u$.

□

Remark 2 *Theorem 2 and Corollary 3 are the consequences of FIFO and the use of the communication buffer (CB). In fact, if we use the traditional methods that each robot only sends the current sensor measurement to neighboring robots without the use of CB, it can happen that two robots may never exchange their sensor measurements, even there exists a path connecting them. The condition of frequently jointly strongly connectedness is also crucial for guaranteeing the consistency of the FIFO-based distributed Bayesian filter, as shown in Section 3.4.5.*

3.4.2 The FIFO-DBF Algorithm

The generic DBF is not directly applicable to time-varying interaction topologies. This is because changing topologies can cause intermittent and out-of-sequence arrival of measurements from different robots, giving rise to the OOSM problem. One possible solution is to ignore all measurements that are out of the temporal order. This is undesirable since this will cause significant information loss. Another possible remedy is to fuse all measurements by running the filtering algorithm from the beginning at each time step. However, this solution causes excessive

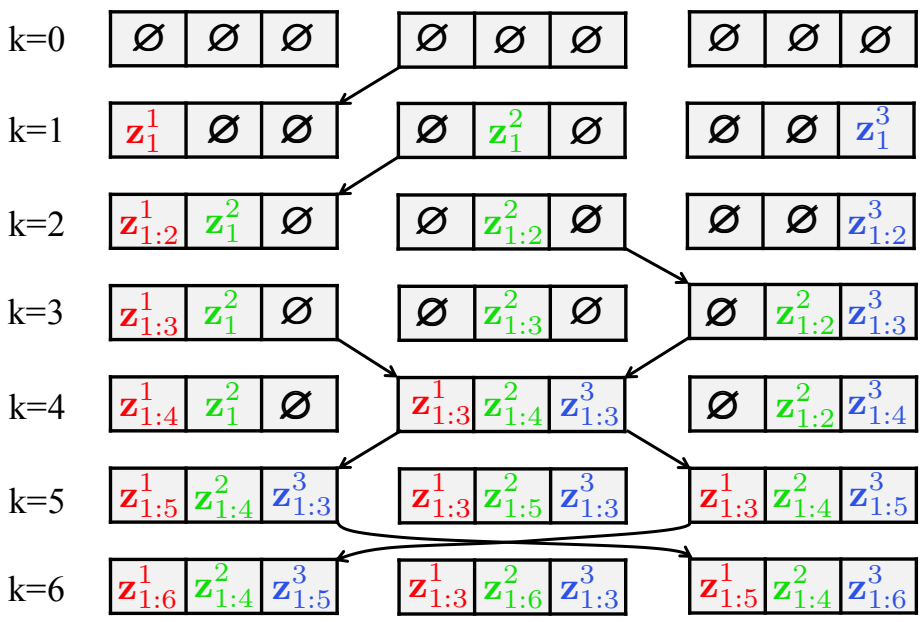


Figure 3.9: Example of FIFO with three robots under dynamically changing interaction topologies. The arrows represent a directed communication link between two robots. ∅ denotes the empty set. For the purpose of clarity, we only show measurements, not the states, in the CB.

computational burden. To avoid both OOSM problem and unnecessary computational complexity, similar to LIFO-DBF, we use the stored PDF, $P^i_{sto}(X_t)$, that is updated from the i^{th} robot's initial PDF by fusing the state-measurement pairs of *all* robots up to a certain time $t \leq k$. The choice of t is described in Section 3.4.3. The individual PDF, $P^i_{pdf}(X_k|\mathcal{Z}^i_{1:k})$, is then computed by fusing the measurements from time $t+1$ to k in the CB into $P^i_{sto}(X_t)$, running the Bayesian filter (Eq. (3.3) and (3.4)). Note that initially, $P^i_{sto}(X_0) = P^i_{pdf}(X_0)$.

The **FIFO-DBF algorithm** is stated in Algorithm 5. Each robot runs FIFO-DBF after its CB is updated in the Updating Step in Algorithm 4. At the beginning, we assign the stored PDF to a temporary PDF, which will then be updated by sequentially fusing measurements in the CB to obtain the individual PDF. It should be noted that, when the robot's CB contains all robots' state-measurement pairs from t to ξ, the temporary PDF corresponding to time ξ is assigned as the new stored PDF. Figure 3.10 illustrates the FIFO-DBF procedure for the 1^{st} robot as an example. It can be noticed that, the purpose of using the stored PDF is to avoid running the Bayesian filtering from the initial PDF at every time step. Since the stored PDF has incorporated all robots' measurements up to time step t, the information loss is prevented. We point out that the time t of each robot's stored PDF can be different from others. The stored PDF is saved locally by each robot and not transmitted to others. FIFO-DBF is able to avoid the OOSM issue since all measurements are fused in the correct temporal order.

Algorithm 5 FIFO-DBF Algorithm

For i^{th} robot at k^{th} step ($\forall i \in V$):

(1) Initialize a *temporary PDF* by assigning the stored individual PDF to it:

$$P^i_{tmp}(X_t) = P^i_{sto}(X_t),$$

where

$$P^i_{sto}(X_t) = P^i_{pdf}(X_t|z^1_{1:t}, \ldots, z^N_{1:t}).$$

(2) For $\xi = t+1$ to k, iteratively repeat two steps of Bayesian filtering:

(2.1) Prediction

$$P^{pre}_{tmp}(X_\xi) = \int_S P(X_\xi|X_{\xi-1}) P^i_{tmp}(X_{\xi-1}) dX_{\xi-1}.$$

(2.2) Updating

$$P^i_{tmp}(X_\xi) = K_\xi P^{pre}_{tmp}(X_\xi) P(\mathcal{Z}^i_\xi|X_\xi),$$
$$K_\xi = \left[\int_S P^{pre}_{tmp}(X_\xi) P(\mathcal{Z}^i_\xi|X_\xi) dX_\xi\right]^{-1}.$$

(2.3) If $z^j_\xi \neq \emptyset$ for $\forall j \in V$, update the stored PDF:

$$P^i_{sto}(X_\xi) = P^i_{tmp}(X_\xi).$$

(3) The individual PDF of i^{th} robot at time k is $P^i_{pdf}(X_k|\mathcal{Z}^i_{1:k}) = P^i_{tmp}(X_k)$.

3.4.3 Track Lists for Trimming CBs

The size of CBs can keep increasing as measurements cumulate over time. The use of the stored PDF has made it feasible to trim excessive state-measurement pairs from the CBs. To avoid information loss, a state-measurement pair can only be trimmed from a robot's CB when *all* robots have received it. We design the *track list* (TL) for each robot to keep track of all robots' reception of other robots' measurements. We first define a binary term $q^{jl}_{k^{ij}}$ ($\forall i, j, l \in V$): $q^{jl}_{k^{ij}} = 1$ if the i^{th} robot knows that the j^{th} robot has received the state-measurement pair of the l^{th} robot of time k^{ij}, $\left[x^l_{k^{ij}}, z^l_{k^{ij}}\right]$; $q^{jl}_{k^{ij}} = 0$ if the i^{th} robot cannot determine whether $\left[x^l_{k^{ij}}, z^l_{k^{ij}}\right]$ has been received by the j^{th} robot. Therefore it can happen that $\left[x^l_{k^{ij}}, z^l_{k^{ij}}\right]$ has been received by the j^{th} robot but the i^{th} robot does not know this and thus $q^{jl}_{k^{ij}} = 0$. Now we define the i^{th} robot's track list as $Q^i_k = \left[\mathbf{q}^{i1}_{k^{i1}}, \ldots, \mathbf{q}^{iN}_{k^{iN}}\right]^T$ ($\forall i \in V$), which is a $N \times (N+1)$ binary matrix with $\mathbf{q}^{ij}_{k^{ij}} = \left[q^{j1}_{k^{ij}}, \ldots, q^{jN}_{k^{ij}}, k^{ij}\right]^T$ ($j \in V$). The last column of Q^i_k, $\left[k^{i1}, \ldots, k^{iN}\right]$, corresponds to measurement times.

The exchange and updating of TLs are described in Algorithm 4, with the updating details

[9] '$\vee$' is the notation of the logical 'OR' operator.

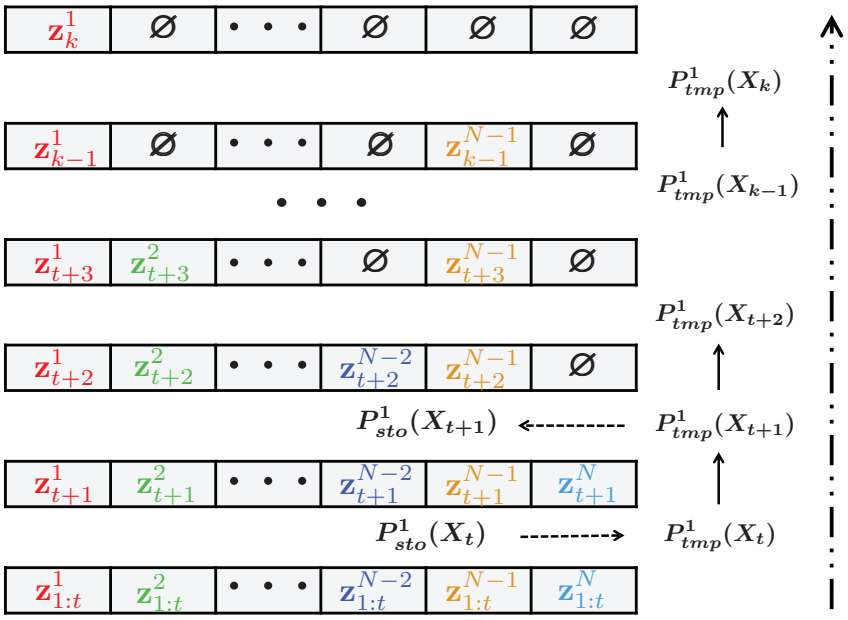


Figure 3.10: Example of FIFO-DBF for the 1st robot at time k. Only the measurement (not the state) is shown in the figure. The robot first calculates $P^1_{tmp}(X_{t+1})$. Since the robot has received all robots' measurements of $t+1$, the $P^1_{tmp}(X_{t+1})$ is assigned as the new stored PDF. The dashed arrow on the right shows the order to fuse measurements in the CB.

Algorithm 6 Updating TLs

Consider updating the i^{th} robot's TL, $\mathcal{Q}^i_k$, using the received r^{th} robot's TL, $\mathcal{Q}^r_{k-1}(r \in \mathcal{N}^{\text{in}}_i(G_{k-1}))$.
(1) Update the i^{th} row of $\mathcal{Q}^i_k$, using $\mathcal{B}^i_k$:

- choose k^{ii} as the minimum integer that satisfies the following conditions: (a) $\exists l \in V$ s.t. $\left[x^l_{k^{ii}}, z^l_{k^{ii}}\right] \notin \mathcal{B}^i_k$ and (b) $k^{ii} \geq t_m - 1$, where t_m is the minimum time of state-measurement pairs in $\mathcal{B}^i_k$.

(2) Update other rows of $\mathcal{Q}^i_k$: $\forall j \in V \setminus \{i\}$

- if $k^{ij} > k^{rj}$, keep current $\mathbf{q}^{ij}_{k^{ij}}$;
- if $k^{ij} = k^{rj}$, $\mathbf{q}^{ij}_{k^{ij}} = \mathbf{q}^{ij}_{k^{ij}} \vee {}^9\mathbf{q}^{rj}_{k^{rj}}$;
- if $k^{ij} < k^{rj}$, $\mathbf{q}^{ij}_{k^{ij}} = \mathbf{q}^{rj}_{k^{rj}}$ and $k^{ij} = k^{rj}$.

presented in Algorithm 6. For the i^{th} robot, it updates the i^{th} row of its TL matrix using the entries of its CB, and updates other rows of the TL using the received TLs from inbound neighbors. The updating rule guarantees that, if the last term of the j^{th} row is k^{ij}, the i^{th} robot is ensured that every robot has received all robots' state-measurement pairs of times earlier than k^{ij}. Figure 3.11 shows the updating of each robot's TL using Algorithm 6. We can use TLs to trim CBs, which is described in Algorithm 7. In the example of Figure 3.11, the 1st and 3rd robot's CB will be trimmed at $k = 6$ and the trimmed state-measurement pairs correspond to times 1, 2, and 3.

The use of TLs can avoid the excessive size of CBs and guarantee that trimming the CBs will not lose any information; the trimmed measurements have been encoded into the stored PDF. The following theorem formalizes this property.

Theorem 3 *Each robot's estimation result using the trimmed CB is the same as that using the non-trimmed CB.*

Proof: Consider the i^{th} robot. Let $k_m^i = \min_j k^{ij}$. Trimming $\mathcal{B}_k^i$ happens when all entries in $\mathcal{Q}_k^i$ corresponding to time k_m^i equal 1. This indicates that each robot has received all robots' state-measurement pairs of time k_m^i. A robot has either stored the pairs in its CB or already fused them to obtain the stored PDF. In both cases, such pairs are no longer needed to be transmitted. Therefore, it causes no loss to trim theses measurements. □

The following theorem describes when CBs get trimmed, and it provides an upper bound of the communication burden that FIFO-DBF will incur. A detailed complexity analysis of FIFO-DBF is presented in Section 3.4.4. Consider trimming all the state-measurement pairs of time t in the i^{th} robot's CB. Let $k_t^{lj} (> t)$ be the first time that the l^{th} robot communicates to the j^{th} robot in the time interval (t, ∞). Define $\tilde{k}_t^j = \max_l k_t^{lj}$, which is the time that the j^{th} robot receives all other robots' measurements of t. Similarly, let $k_t^{ji} (> \tilde{k}_t^j)$ be the first time that the j^{th} robot communicates to the i^{th} robot in the time interval $(\tilde{k}_t^j, \infty)$ and define $\tilde{k}_t^i = \max_j k_t^{ji}$. The following theorem gives the time when the i^{th} robot $(\forall i \in V)$ trims all state-measurement pairs of time t in its own CB.

Theorem 4 *The i^{th} robot trims $\left\{\left[x_t^l, z_t^l\right] \ (\forall l \in V)\right\}$ from its CB at the time $\tilde{k}_t^i$.*

Proof: The i^{th} robot can trim $\left\{\left[x_t^l, z_t^l\right] \ (\forall l \in V)\right\}$ only when it is sure that all other robots have also received these state-measurement pairs. This happens at $\tilde{k}_t^i$ and thus $\tilde{k}_t^i$ is the time when the trim occurs. □

Corollary 4 *Under the frequently jointly strongly connectedness condition, the size of any robot's CB is no greater than $2N(N-1)T_u$.*

Proof: We consider an arbitrary i^{th} $(i \in V)$ robot. According to Theorem 2, a robot can communicate to any other robot within NT_u steps. Therefore, $\tilde{k}_t^i \leq 2NT_u$, since it first requires each robot communicate to all other robots and then each robot communicate to the i^{th} robot. This implies

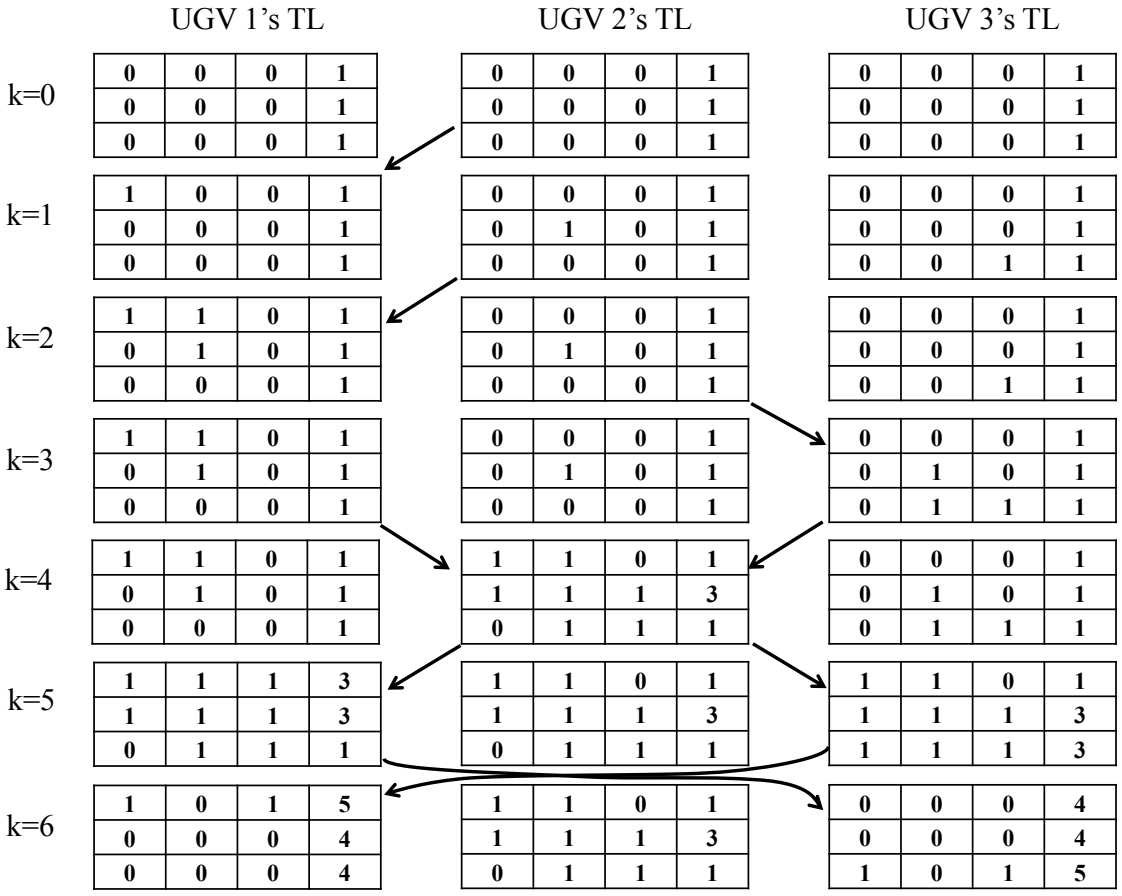


Figure 3.11: Example of updating TLs. For the 1st robot's TL, the j^{th} ($j \in V$) entry on the i^{th} ($i \in V$) row represents this robot's knowledge about whether the i^{th} robot has received the j^{th} robot's state-measurement pair of time k^{1i}, where k^{1i} is the last entry of the i^{th} row. TLs are updated using Algorithm 6.

that, the state-measurement pairs of a certain time of all robots will be trimmed from each robot's CB within $2NT_u$ steps.

The maximum size the of CB occurs when the state-measurement pairs of a certain time from all but one robot are saved in the i^{th} robot's CB. Therefore, the size of any robot's CB is no greater than $2N(N-1)T_u$.

□

3.4.4 Complexity Analysis of FIFO-DBF

Compared to statistics dissemination, FIFO is usually more communication-efficient for distributed filtering. To be specific, consider a grid representation of the environment with the size $D \times D$. The transmitted data between each pair of robots are the CB and TL of each robot. The size of the CB is upper bounded by $O(N^2T_u)$, according to Corollary 4. On the contrary, the communicated

Algorithm 7 Trimming CBs using TLs

For the i^{th} robot: find the smallest time in $\mathcal{Q}_k^i$: $k_m^i = \min\left\{k^{i1}, \ldots, k^{iN}\right\}$.

1. Remove state-measurement pairs in $\mathcal{B}_k^i$ that corresponds to measurement times earlier than k_m^i, i.e., $\mathcal{B}_k^i = \mathcal{B}_k^i \setminus \left\{\left[x_t^l, z_t^l\right]\right\}$, $\forall t < k_m^i, \forall l \in V$.
2. If entries associated with time k_m^i in $\mathcal{Q}_k^i$ are 1's, then
   a) set these entries to be 0.
   b) update the i^{th} row of $\mathcal{Q}_k^i$ using the current CB, i.e., $q_{k^{ii}}^{il} = 1$ if $\left[x_{k^{ii}}^l, z_{k^{ii}}^l\right] \in \mathcal{B}_k^i$, $\forall l \in V$.
   c) remove all corresponding state-measurement pairs in $\mathcal{B}_k^i$, i.e., $\mathcal{B}_k^i = \mathcal{B}_k^i \setminus \left\{\left[x_{k_m^i}^l, z_{k_m^i}^l\right]\right\}$, $\forall l \in V$.
   d) $k_m^i \leftarrow k_m^i + 1$.

data of a statistics dissemination approach that transmits unparameterized posterior distributions or likelihood functions is $O(D^2)$. In applications such as the target localization, D is generally much larger than N. Besides, the consensus filter usually requires multiple rounds to arrive at consensual results. Therefore, when T_u is not comparable to D^2, the FIFO protocol requires much less communication burden.

It is worth noting that each robot needs to store an individual PDF and a stored PDF, each of which has size $O(D^2)$. In addition, each robot needs to keep the CB and TL. This is generally larger than that of statistics dissemination-based methods, which only stores the individual PDF. Therefore, the FIFO-DBF sacrifices the local memory for saving the communication resource. This is actually desirable for real applications as local memory of vehicles is usually abundant compared to the limited bandwidth for communication.

Remark 3 *Under certain interaction topologies, CBs can grow to undesirable sizes and cause excessive communication burden if the trim cannot happen frequently. In this case, we can use a time window to constrain the measurements that are saved in CBs. This will cause information loss to the measurements. However, with a decently long time window, FIFO-DBF can still effectively estimate the target position.*

3.4.5 Consistency Proof

This section proves the consistency of the *maximum a posteriori* (MAP) estimator of FIFO-DBF under unbiased sensors (sensors without offset) for static robots. Similar to Section 3.3.4, we assume that S is a finite set (e.g. a finely discretized field) and the target position can be uniquely

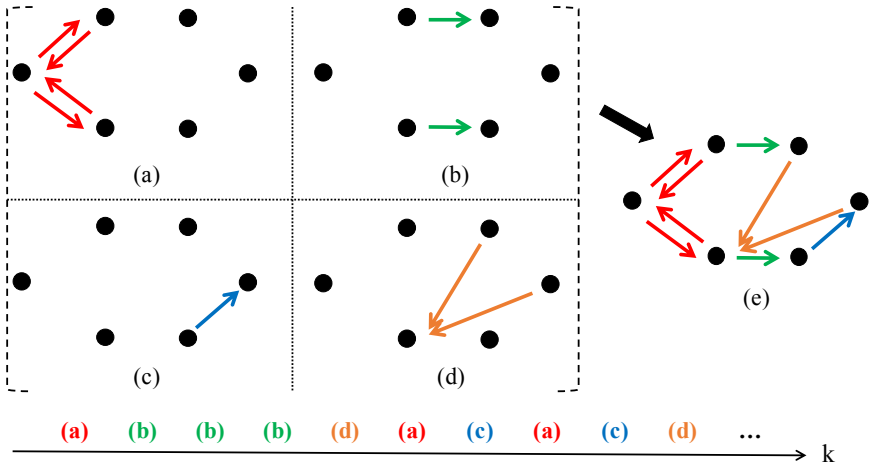


Figure 3.12: The dynamically changing interaction topologies used in the simulation: (a)-(d) four types of topologies; (e) the union of these topologies is jointly strongly connected. The bottom axis shows a randomly generated sequence of topologies that satisfy the frequently jointly strongly connectedness condition.

determined by the multi-robot network with proper placement (i.e., excluding the special case of ghost targets [122]).

The consistency of FIFO-DBF for static robots is stated as follows:

Theorem 5 *Assume the robots are static and the sensors are unbiased. If the network of N robots is frequently jointly strongly connected, then the MAP estimator of target position converges in probability to the true position of the target using FIFO-DBF, i.e.,*

$$\lim_{k\to\infty} P(X_k^{MAP,i} = x^g) = 1, \ i \in V,$$

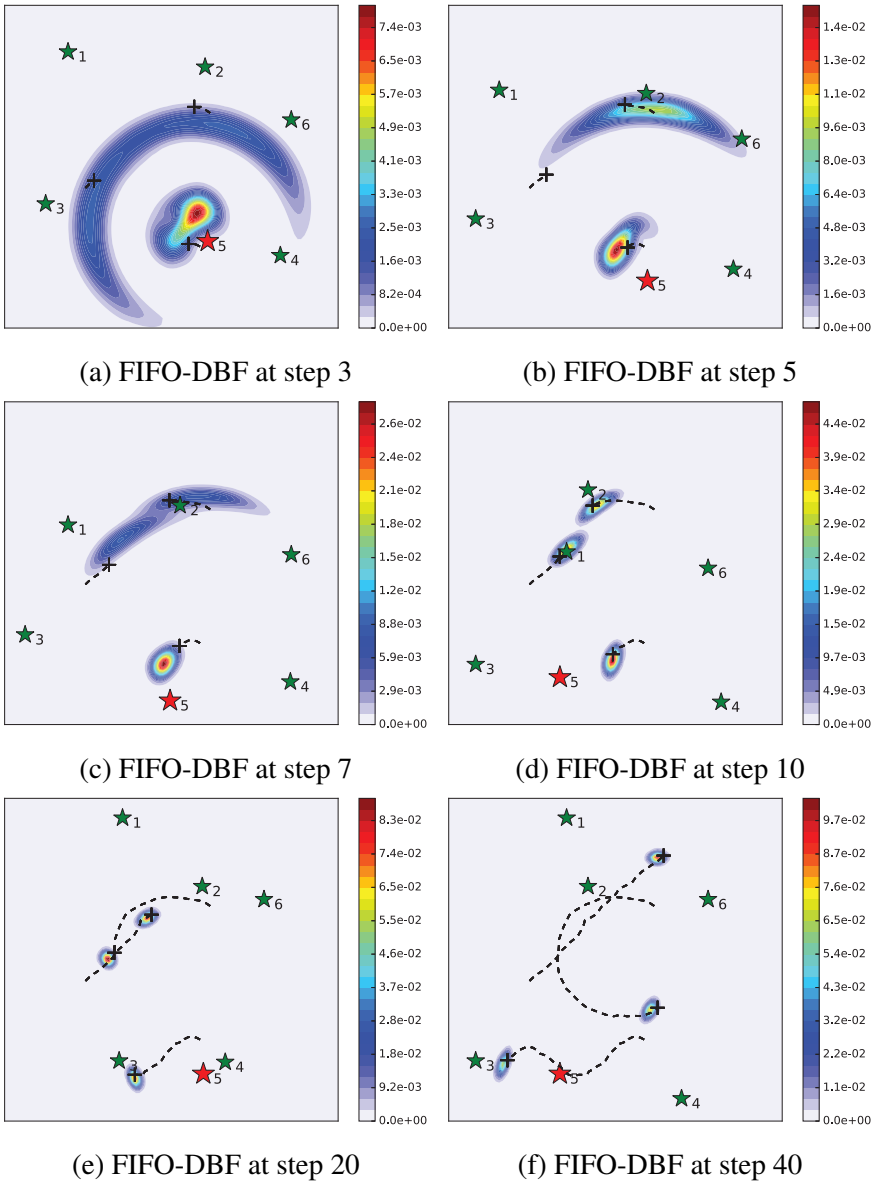
where

$$X_k^{MAP,i} = \arg\max_X P^i_{pdf}(X|\mathcal{Z}^i_{1:k}).$$

Proof: The main body of the proof is similar to that of Theorem 1. The only difference is that to obtain Eq. (3.7), we no longer have the result of Corollary 2 in FIFO case. Instead, we use the finite delay of measurement arrival from Corollary 3, i.e., $k - NT_u \leq |\mathcal{K}_k^{ij}| \leq k$, where $|\cdot|$ is the set cardinality. This allows the derivation of Eq. (3.7) under FIFO-DBF. For the rest of proof, refer to the proof of Theorem 1. □

3.4.6 Simulation

We conduct a simulation that uses a team of six robots to localize three moving targets. Every robot maintains three individual PDFs, each corresponding to a target. At each time step, a robot's sensor can measurement the positions of three targets. We assume that the robots know the association

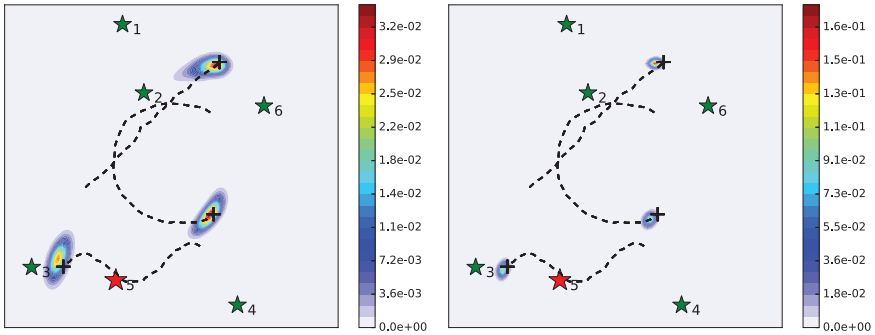


(a) FIFO-DBF at step 3 (b) FIFO-DBF at step 5

(c) FIFO-DBF at step 7 (d) FIFO-DBF at step 10

(e) FIFO-DBF at step 20 (f) FIFO-DBF at step 40

Figure 3.13: Evolution of the 1st robot's target estimation using FIFO-DBF. The colorful background represents the sum of the individual PDF of three targets.



(a) CbDF at Step 40 (b) CF at Step 40

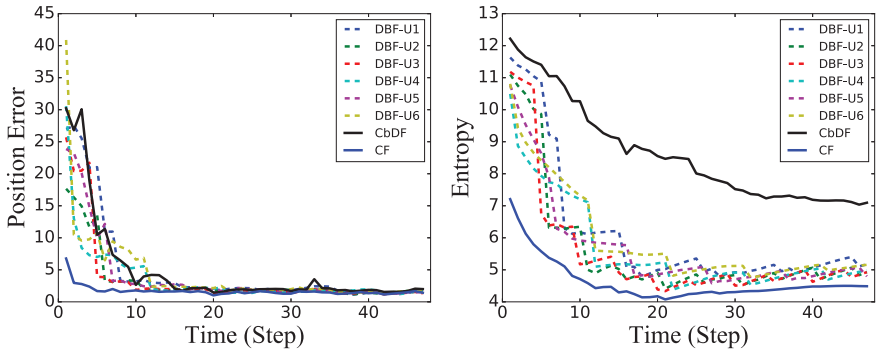
Figure 3.14: The 1st robot's target estimation using (a) CbDF and (b) CF. The estimation uncertainty remains large for CbDF.

between the measurement and the corresponding target. The targets have different motion models, including the linear motion (target 1), sinusoidal motion (target 2), and circular motion (target 3). Three of the robots have range-only sensors and the other three robots have bearing-only sensors. The interaction topology of the robots is time-varying and consists of four types, as shown in Figure 3.12(a)-(d). A randomly generated sequence of topologies is used (Figure 3.12f). It can be noticed that, the interaction topology is frequently jointly strongly connected when all four types appears repeatedly (Figure 3.12e). Ten test trials are used, with the randomly generated initial positions of robots and targets. There exists different methods to implement a Bayesian filter, including the histogram filter and the particle filter [158]. The histogram filter is easy to implement and can keep track of the probability mass over the whole field, but can be computationally heavy for large fields. The particle filter, on the contrary, is advantageous when the field is very large, but can introduce inaccuracy due to particle deprivation [158]. We use both methods to implement the Bayesian filter in the simulation, and their results are very similar. For the purpose of clarity, we only include the results from the histogram filter here.

We compare FIFO-DBF with two commonly adopted approaches in multi-agent filtering: the consensus-based distributed filter (CbDF) [132] and the centralized filter (CF) [166]. The CbDF requires robots to continually exchange their individual PDFs with neighbors, computing the average of its own and the received PDFs. Multiple rounds of communication and averaging are needed at each step to ensure the convergence of robots' individual PDFs. The CF assumes a central unit that can constantly receive and fuse all robots' latest measurements into a single PDF.

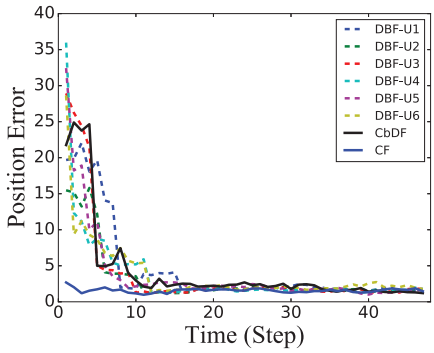
Figure 3.13 shows the simulation results of a specific trial. The sum of the 1st robot's three individual PDFs are shown in the figures. They show that the FIFO-DBF can successfully localize and track moving target's positions and effectively reduce the estimation uncertainty, which is similar to the performance of the CF (Figure 3.14b). On the contrary, CbDF is less effective in reducing the estimation uncertainty (Figure 3.14a).

We quantitatively compare the three filters in terms of the estimation error and entropy of the uncertainty. The average of the estimation error and entropy of each target across ten trials are shown in Figure 3.15. It can be noticed that, the CF achieves the most accurate position

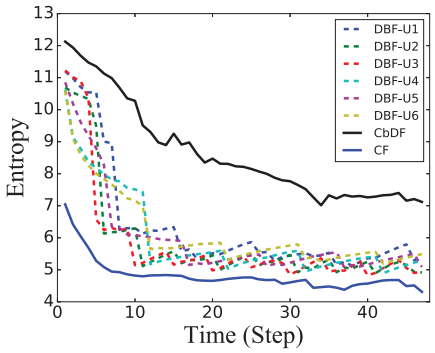


(a) Position Error of Target 1

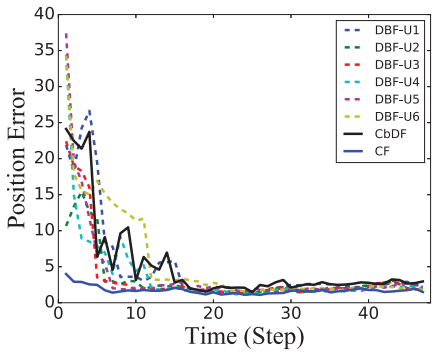
(b) Entropy of Target 1



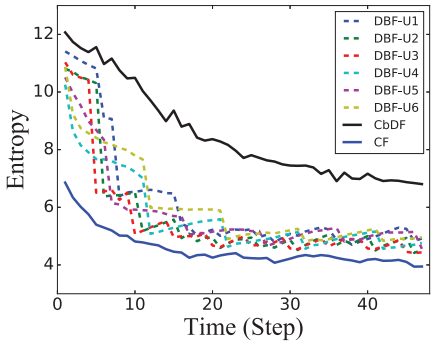
(c) Position Error of Target 2



(d) Entropy of Target 2



(e) Position Error of Target 3



(f) Entropy of Target 3

Figure 3.15: The average estimation error (a)(c)(e) and the average entropy (b)(d)(f) of ten trials using different filtering approaches. The dotted lines correspond to the results of FIFO-DBF by the six robots ('U1'-'U6' in the legends).

estimation and fastest entropy reduction. This is an expected result since the CF utilizes all sensor measurements. The FIFO-DBF achieves similar results as the CF asymptotically. This is a very interesting results, since FIFO-DBF only communicates with neighboring robots and have a subset of other robots' measurements. The CbDF achieves similar position estimation performance as the CF and FIFO-DBF. However, it fails to effectively reduce the estimation entropy. This is because that, the linear combination of PDFs used in the CbDF does not follow the nonlinear nature of Bayesian filtering, thus information is loss during the combination. The FIFO-DBF, on the other hand, rigorously follows the procedure of Bayesian filtering, and therefore achieves better performance. Besides, CbDF requires multiple rounds of exchanging individual PDFs, which incurs much higher communication burden than FIFO-DBF at each time step. Therefore, FIFO-DBF is more preferable than CbDF.

3.5 Summary

This chapter presents two measurement-dissemination-based distributed Bayesian filters (DBFs) for a network of multiple robots. For time-invariant networks, we propose the Latest-In-and-Full-Out (LIFO) method for data exchange among robots. LIFO ensures the dissemination of the sensor measurements among robots and significantly reduces the amount of communication data in networks. Depending on whether the target is static or moving, two LIFO-DBFs are proposed. The consistency property using LIFO-DBF has been proved. An indoor experiment was conducted to validate the effectiveness of LIFO-DBF for distributed filtering.

For time-variant networks, we propose the Full-In-and-Full-Out (FIFO) method. We identify the frequently jointly strongly connectedness condition on the network connectivity to guarantee the dissemination of the sensor measurements among robots. A track list is designed to cut down the communication burden of FIFO while avoiding information loss. Compared to the statistics-dissemination-based approaches, FIFO significantly reduces the communication data for the network. A FIFO-based DBF is proposed and its consistency property is proved. Simulation results show that the proposed LIFO-DBF and FIFO-DBF achieve comparable performance as centralized filters and better results than consensus-based approaches.

Chapter 4

Path Planning for Target Search and Tracking

4.1 Introduction

Using autonomous robots for target search and tracking has received wide interests in recent years due to potential applications on environment monitoring, surveillance, and disaster response. In spite of achievements so far (Section 2.3.3), several issues remain that hinder the practical use of this technique: (1) Common methods have either used one-step prediction to compute a myopic trajectory or solved a high-dimensional optimization problem. The former results in less effective search due to short planning horizon while the latter involves numerical integration over large state space that usually results in high computational complexity. (2) The collision avoidance in dynamic environments that contain static and moving obstacles is less investigated. The negligence of collision avoidance is generally acceptable for aerial and marine search, but not suitable for ground environment, which may be crowded with pedestrians, motor vehicles, buildings, and other inaccessible areas. (3) Most of works did not explicitly take into account the effects of limited sensing domain of sensors in the planning process.

The dissertation work in this chapter presents MPC-based probabilistic planning approaches for non-myopic informative path planning of an autonomous robot to search for and localize/track a static/moving target. MPC is naturally suitable for this because the estimation of target position will constantly get updated as new sensor measurements are obtained, and MPC can utilize this updated target information to re-generate a path at each time instance. The main contributions of the work include: (1) For the case that both the target motion model and sensor measurement model are linear/affine (Section 4.2), the dissertation work develops an MPC-based planner that explicitly considers the limited sensing domain in the planning process. A sequential programming approach is proposed for generating the robot's trajectory, which reduces the computational complexity caused by the limited sensing domain. (2) In the case that the sensor uses a binary model (Section 4.3), this work proposes an analytic objective function that alleviates the computational burden compared to traditionally used numerical integral approaches. A barrier function approach is used to enforce collision avoidance between the robot and obstacles in the environment. In the

following sections, each scenario will first be defined and the proposed planning approaches are then presented.

4.2 Path Planning with A Linear Sensor[1]

We consider a two-dimensional planar space, as shown in Figure 4.1, that contains a moving target (blue cylinder) and an autonomous mobile robot (red wheeled robot). The robot is equipped with a sensor that has limited sensing domain to measure the target position. It is assumed that the sensor has the same state as the robot (i.e. position and heading angle). The target position is unknown a priori and the robot needs to autonomously search for the target. Once having detected the target, the robot will need to continue tracking it by keeping the target in its sensor's sensing domain. As shown in the figure, the target position comes with large initial uncertainty due to the lack of prior information. However, when the target is within the sensing domain, the uncertainty is significantly reduced and the robot will then keep tracking the target. The target motion model is assumed to be linear (Eq. (2.2)) and the robot motion model uses the unicycle model in Eq. (2.12).

A particularly relevant work was done by Patil et al. [134], who considered the path planning in Gaussian belief space and proposed a sequential planning algorithm in MPC framework to deal with the discontinuity of sensing domain. The work in this section is different from that work in the way the limited sensing domain is modeled and the strategy for solving the MPC problem. To be specific, a geometric representation of sensing domain is used to define its boundaries. In addition, a two-step optimization structure is proposed that successively improves the planned path to reduce the uncertainty of the target. Such approach can alleviate the computational burden of solving the MPC problem.

4.2.1 Sensor Modeling

Modeling Sensing Domain

Common sensors, such as cameras and radars, have limited sensing domain that are bounded in both sensing range (distance) and angle of view [181]. We approximate the sensing domain as a sector (Figure 4.2), which can be represented as $\mathcal{F}_k = \left\{[x^g_{1,k}, x^g_{2,k}] \in \mathbb{R}^2 | \, \|\mathbf{q}\|_2 \leq r, \angle\mathbf{q} \in [\theta_1, \theta_2]\right\}$, where $\mathbf{q} = [x^g_{1,k} - x^r_{1,k}, x^g_{2,k} - x^r_{2,k}]$. It should be noted that, though we use a sector-shaped sensing domain model in this work, the presented path planning method can also apply to other geometries of sensor's sensing domain, such as the lobe or cone shape.

Sensor Measurement Model

We assume a linear measurement model as shown in Eq. (2.4). Since the target position is unknown a-priori, the target can be outside of the sensing domain and results in missing measurement. To tackle this problem, we adopt the measurement model from Sinopoli et al. [152], which provides

[1]This section is based on the work published in [111].

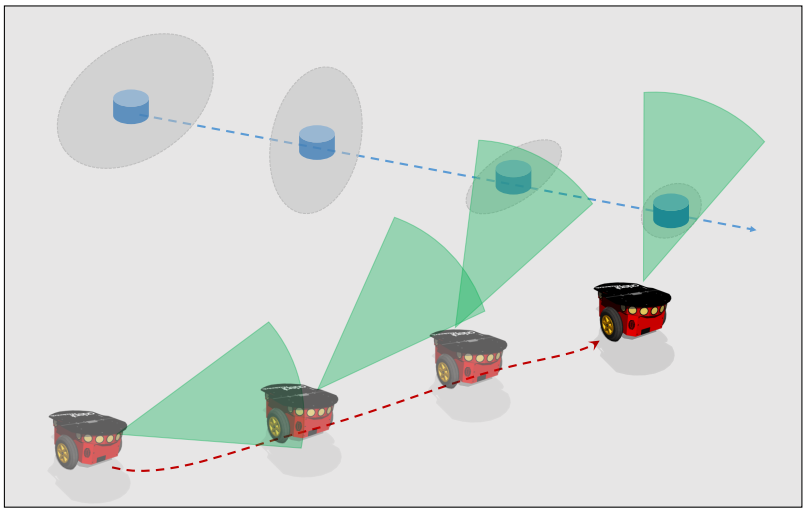


Figure 4.1: Scenario of target search and tracking. Trajectories of the robot and the target are shown as the red and blue dashed lines. The gray circles correspond to the uncertainty of the estimated target position. The green sector shows the sensor's limited sensing domain.

a unified model for handling intermittent measurements:

$$z_k = Cx_k^g + w_k,\ w_k \sim \begin{cases} \mathcal{N}(0, R) & \text{if } \gamma_k = 1 \\ \mathcal{N}(0, \sigma^2 I) & \text{if } \gamma_k = 0 \end{cases}, \tag{4.1}$$

where $C \in \mathbb{R}^{2\times 2}$ is the measurement matrix and $w_k \in \mathbb{R}^2$ is a zero-mean Gaussian noise. γ_k is a binary random variable denoting whether a measurement is received ($\gamma_k = 1$) or not ($\gamma_k = 0$) at time step k. Different from the measurement noise in traditional Kalman filter, covariance of w_k now depends on whether a measurement is obtained. The receiving of a measurement corresponds to a finite covariance $R \succeq 0$ and the absence of measurement corresponds to the limiting case of $\sigma \to \infty$.

To adapt Eq. (4.1) for handling limited sensing domain, we define

$$\gamma_k = \mathbb{1}_{\{x_k^g \in \mathcal{F}_k\}} \tag{4.2}$$

to be an indicator function that reflects whether the target is within the sensing domain (a measurement is thus received) or outside of the sensing domain (the measurement is therefore missing). This formulation will facilitate the development of a unified MPC that explicitly handles intermittent measurements caused by limited sensing domain, which is to be described in following sections.

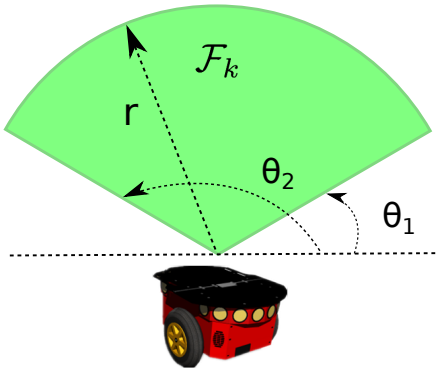


Figure 4.2: Illustration of sensor's sensing domain $\mathcal{F}_k$. The sensing range is r and the angle of view is $[\theta_1, \theta_2]$.

4.2.2 Kalman Filter with Limited Sensing Domain

Since limited sensing domain can cause intermittent measurements, we utilize the discrete-time Kalman filter from [152] to consider this effect, which is defined as

$$\hat{x}^g_{k+1|k} = f^g(\hat{x}^g_{k|k}) \tag{4.3a}$$
$$P_{k+1|k} = AP_{k|k}A' + Q \tag{4.3b}$$
$$K_{k+1} = P_{k+1|k}C(CP_{k+1|k}C' + R)^{-1} \tag{4.3c}$$
$$\hat{x}^g_{k+1|k+1} = \hat{x}^g_{k+1|k} + \gamma_{k+1}K_{k+1}(z_{k+1} - h(\hat{x}^g_{k+1|k}; x^r_{k+1|k})) \tag{4.3d}$$
$$P_{k+1|k+1} = P_{k+1|k} - \gamma_{k+1}K_{k+1}CP_{k+1|k}. \tag{4.3e}$$

Note that $\hat{x}^g_{k+1|k}$ and $P_{k+1|k}$ are functions of γ_{k+1} now and thus depend on the state of both the robot and target.

4.2.3 Path Planning for Target Search and Tracking

We formulate the path planning problem using the model predictive control framework. For the clarity of description, we use $x^r_{k+i|k}$ to represent the planned robot state at time $k+i$ with the current time step being k. Similarly, $\hat{x}^g_{k+i|k}$ represents the predicted target state at time $k+i$. For notational simplicity, we also define $b_{k+i+1|k} = [\hat{x}^g_{k+i+1|k+i}, P_{k+i+1|k+i}, K_{k+i+1}, \hat{x}^g_{k+i+1|k+i+1}, P_{k+i+1|k+i+1}]^T$ $(i = 0, 1, 2, \dots)$ and let $b_{k+i+1|k} = g(b_{k+i|k}, u^r_{k+i})$ represent the Kalman filter defined in Eq. (4.3), called the *filter dynamics*. Note that in the filter dynamics the predicted sensor measurement is used since the actual measurement at $k+i$ is unknown when the current time is k. Here, the maximum a-posteriori (MAP) estimate, $\hat{x}^g_{k+i|k}$, is used as the predicted target position at the future time $k+i$. Therefore the 2nd summand on the right hand side of Eq. (4.3d) is always 0. We use MPC for path planning,

which is formulated as

$$\min_{u^r_{k:k+N-1}} \sum_{i=1}^{N} w_1 tr(P_{k+i|k}) + w_2 \|\hat{x}^g_{k+i|k} - x^r_{k+i|k}\|^2 \tag{4.4a}$$

$$\text{s.t. } x^r_{k+i+1|k} = f^r(x^r_{k+i|k}, u^r_{k+i}), \tag{4.4b}$$

$$b_{k+i+1|k} = g(b_{k+i|k}, u^r_{k+i}), \tag{4.4c}$$

$$x^r_{k+i+1|k} \in \mathcal{X},\ u^r_{k+i} \in \mathcal{U}, \tag{4.4d}$$

$$i = 0, \ldots, N-1,$$

where N is the planning horizon; Eq. (4.4b) is the unicycle model defined in Eq. (2.12); $\mathcal{X}$ and $\mathcal{U}$ represent the feasible sets of robot state and control input, consisting of constraints on robot position, speed, steering rate, and linear acceleration.

To drive the robot to configurations in which the sensor can effectively obtain target information, we choose the cumulative entropy and the distance between the robot and predicted target position as the objective function Eq. (4.4a). Because of the linear Gaussian noise assumption in the target model and measurement model, target uncertainty follows a Gaussian distribution. Therefore, the entropy at k is $H(b_k) = \frac{k}{2}(1 + \ln(2\pi)) + \frac{1}{2}\ln|P_{k|k}|$ ([120]). Minimizing a determinant can incur large computation burden for an optimization problem. Therefore, we utilize the relation $|A|^{\frac{1}{n}} \leq \frac{1}{n} tr(A)$ for a positive definite matrix A and use $tr(P_{k|k})$, the trace of the matrix $P_{k|k}$, to represent the entropy term in the objective function.

The binary variable γ_k (Eq. (4.2)) is a discontinuous function of the robot and target states, which is inconvenient for an optimization problem that usually requires differentiability of functions in it. Therefore we approximate γ_k as a product of two bell-shaped differentiable functions, corresponding to bounds on sensing range and angle:

$$\gamma_k \approx \frac{1}{1 + \alpha_1 \|[x^g_{1,k}, x^g_{2,k}] - [x^r_{1,k}, x^r_{2,k}]\|_2^2} \times \frac{1}{1 + \exp\left\{-\alpha_2(\cos(\theta^r_k - \tilde{\theta}_k) - \cos(\theta_0))\right\}}, \tag{4.5}$$

where $\tilde{\theta}_k = \angle([x^g_{1,k}, x^g_{2,k}] - [x^r_{1,k}, x^r_{2,k}])$ is the direction angle from the sensor position to target position; $\theta_0 = \frac{\theta_2 - \theta_1}{2}$ is half of the sensing angle; α_1 and α_2 are tunning parameters that controls the shape of the function. Eq. (4.5) can be interpreted as follows: when the robot is close to the target, it is more likely that the target can be detected; besides, the closer the target direction aligns with the center direction of the sensor, the higher possibility that the target will get detected. Figure 4.3 shows the value of the approximate γ_k.

Sequential Planning

Directly solving the MPC problem in Eq. (4.4) is time consuming, mainly due to the coupling of robot and target state in γ_k defined in Eq. (4.5). To reduce the computation burden for real-time applications, we make several relaxation to the original MPC problem. To be specific, in

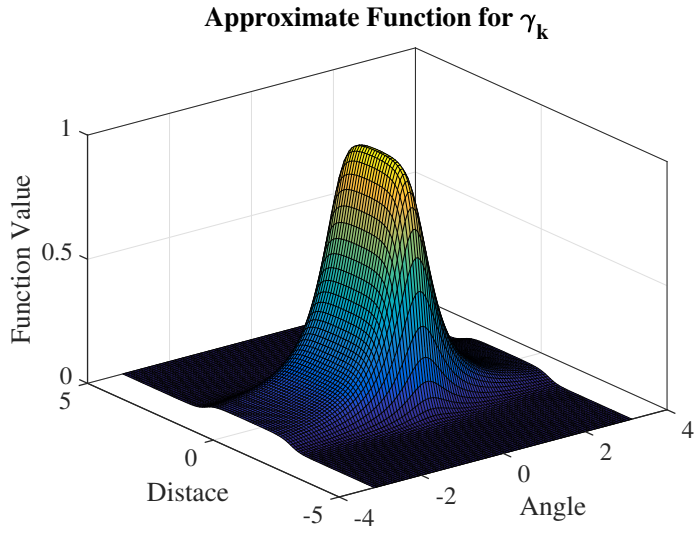


Figure 4.3: Illustration of the bell-shaped function for approximating γ_k. In this example, parameters take the following values: $\tilde{\theta}_k = 0,\ \theta_0 = \pi/3,\ \alpha_1 = 1,\ \alpha_2 = 10$.

the predictive horizon, the target position is assumed to change following its motion model but without noise, starting from the estimated target position (mean value from KF) at the beginning of the prediction. This is equivalent to predicting target position using Eq. (4.3a) but without the updating step (Eq. (4.3d)). This is a reasonable assumption for MPC framework. In fact, because of the stochasticity of target motion and sensor measurement process, it is difficult to accurately predict the evolution of the environment state for a long horizon. The benefit of MPC is that it will takes into account the newly received information about the target and then utilize it to update the path at each time step. This can compensate for inaccuracy caused by using the open-loop prediction in the predictive horizon.

We also devise a two-step process to further reduce the computation burden for the MPC problem. In the first step, we only consider the limited sensing range of the sensor, assuming a full sensing angle. In this step, γ_k is equivalent to the first multiplier in Eq. (4.5). With this simplified γ_k, the MPC computes a reference trajectory of the robot. Utilizing the generated reference trajectory, the heading angle of sensor $\hat{\theta}_k$ at each predicted time can be computed. In the second step, MPC is solved again but with the full expression of γ_k, in which $\hat{\theta}_k$ takes the value from the first step. The benefits of conducting this sequential two-step planning process lies in that computing the reference trajectory in the first step generates the reference heading angle of the robot, which reduces the computation burden when later a full version of MPC is solved.

4.2.4 Simulation

We have tested the algorithm on different scenarios. Each scenario consists of a mobile robot equipped with a camera and a moving target in a $50m \times 50m$ field. The initial position of the target is different in each scenario while the robot starts from the same position. The robot has a maximum

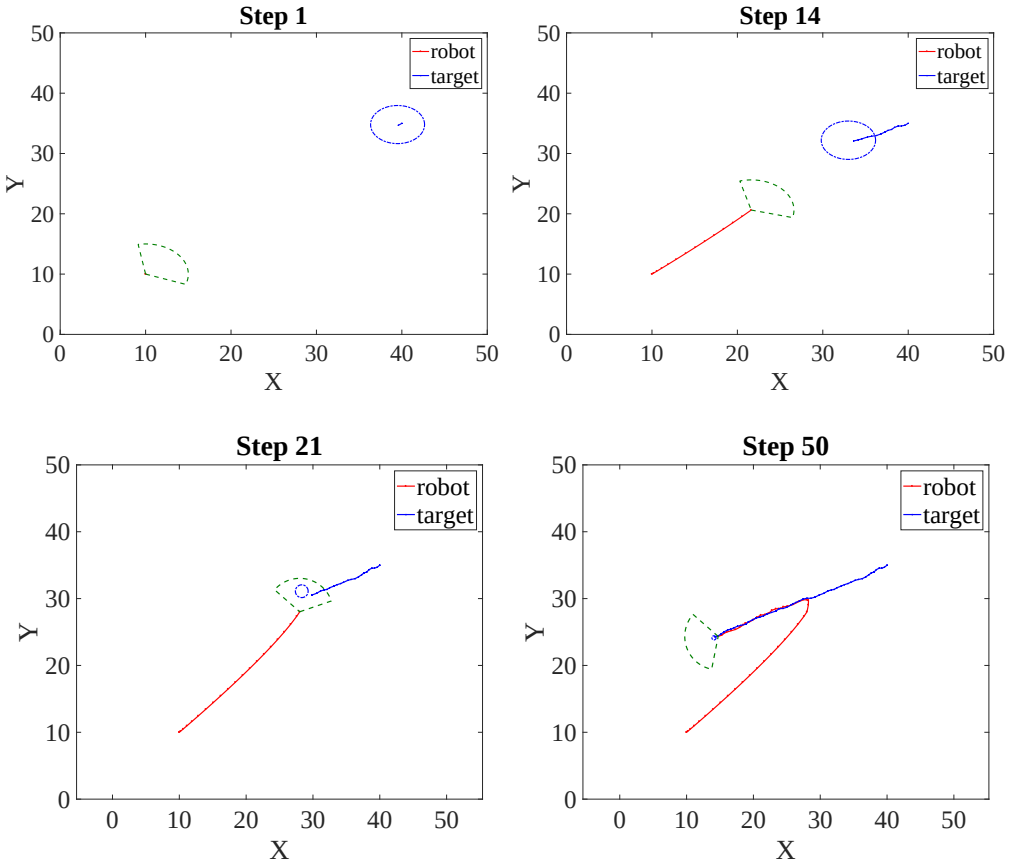


Figure 4.4: Intermediate steps of search and tracking. Green dashed sector represents the sensor's sensing domain and the blue dashed circle show the uncertainty of the target position (based on diagonal elements of $\sqrt{P_{k|k}}$). Red and blue lines correspond to the trajectories of the robot and target.

speed of $3m/s$ with the acceleration range $[-3m/s^2, 1m/s^2]$ and the angular velocity range $[-\frac{\pi}{4}, \frac{\pi}{4}]$. The sensing range $r = 5m$ and the sensing angle is $60°$. Target model A and measurement model C are assumed to be identity matrices, with the noise $R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ and $Q = \begin{bmatrix} 0.01 & 0 \\ 0 & 0.01 \end{bmatrix}$. The value of B varies among different scenarios.

Figure 4.4 shows the intermediate search and tracking steps of a test scenario. At the beginning, the robot starts from the lower left corner and the target is at the upper right corner. The estimated target position is initialized as the target's true position but with a large uncertainty. The robot then moves towards the target by using the control input computed from the MPC problem (Eq. (4.4)). At step 21, the robot successfully detects the target and thus the uncertainty significantly shrinks. The robot then keeps tracking the target and the uncertainty is further reduced, as the figure of Step 50 shows.

Robot and target trajectories in some other test scenarios are shown in Figure 4.5. The lower

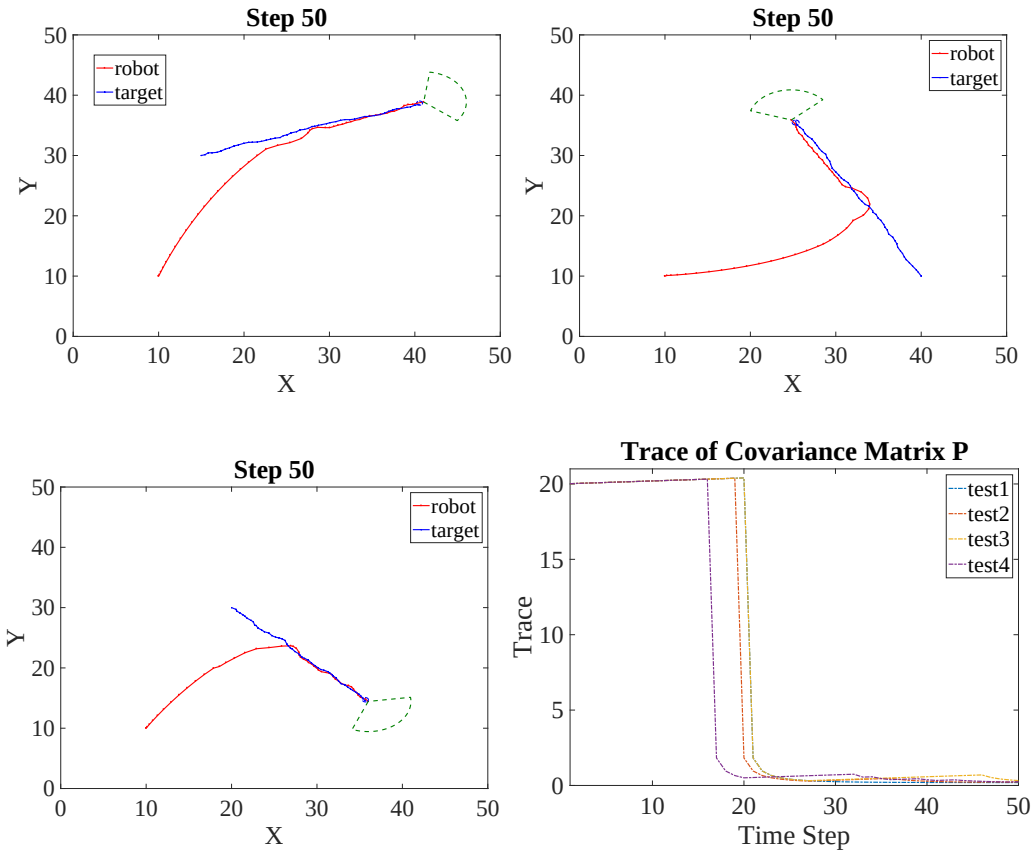


Figure 4.5: First three figures show the final steps of search and tracking mission in three different scenarios. The initial estimated target position is set to be the true position. The last figure shows the trace of $P_{k|k}$ at each time step for all four scenarios.

right figure in Figure 4.5 shows the trace of the covariance matrix of the target position. As expected, uncertainty keeps increasing at the beginning since the target is not found and thus no uncertainty reduction can be conducted by Kalman filter. Once the target is within the sensor's sensing domain, uncertainty drops significantly since Kalman filter is able to use measurements to improve the estimation of the target position. The robot then keeps measuring the target position and the uncertainty continues being reduced. Fluctuation of the trace can also be observed in later part of the search and tracking. This is mainly due to the fact that the target moves in a stochastic manner and therefore there exists some cases that the target falls out of the sensing domain. However, the robot is able to adjust its sensor and keep the target in its sensing domain again. These figures show that the robot successfully detects the target and then keeps tracking it afterwards in all considered cases using the presented approach.

We have also set the initial estimated target position to be different from the true position, which adds more complexity to the problem. Results on three scenarios from Figures 4.4 and 4.5 are presented in Figure 4.6. It shows that the robot is able to find the target and track it in spite of inaccurate initial estimation. However, we also notice that, when the initial estimated position is far

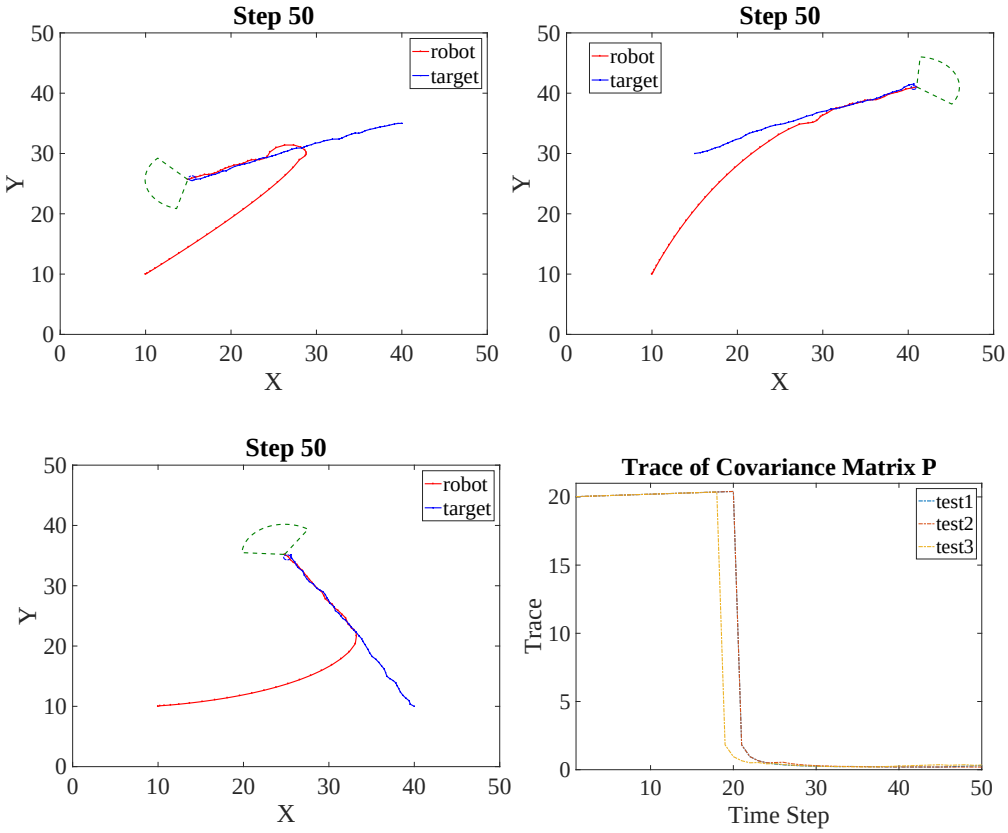


Figure 4.6: First three figures show the final steps in three different scenarios. The initial estimated target position is different from the true position. The last figure shows the trace of $P_{k|k}$ at each time step of these scenarios.

from the true position, it is difficult for the robot to find the target. This is not unexpected since in current formulation the estimated target position is propagated using noise-free system model but without correction if no measurement is received. As [152] indicated, when measurement arrival rate is below certain threshold, the Kalman filter Eq. (4.3) cannot converge.

4.3 Path Planning with A Binary Sensor[2]

The work in this section considers a similar setup as in Section 4.2. However, instead of considering a moving target, the targat is assumed static in this case. As shown in Figure 4.7, there exist several obstacles in the environment. The autonomous ground robot is tasked with localizing the target while avoiding collision with surrounding obstacles. The robot is assumed to be equipped with a binary sensor (Eq. (2.5) and (2.6)) for detecting the stationary target and a perception system for detecting surrounding obstacles. This assumption is reasonable because the target usually gen-

[2]This section is based on the work published in [114].

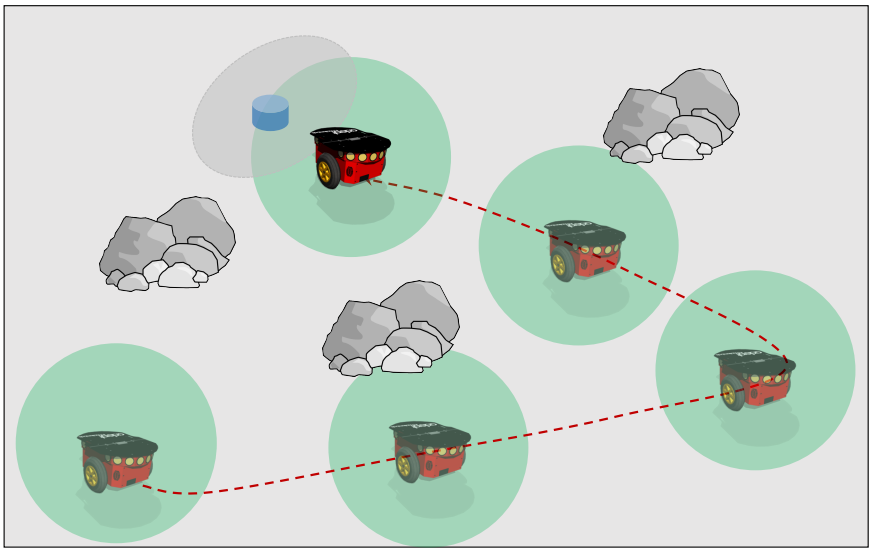


Figure 4.7: Scenario of target search in a dynamic environment. The robot uses the binary sensor to localize the target. It needs to avoid colliding with obstacles (stones).

erates limited information to be accurately positioned. For example, in a rescue mission, a sound level sensor can only report whether there is a survivor within the detection range, while the motion of obstacles can be accurately measured and transmitted to the robot by using technologies such as Global Positioning System (GPS), geographic information system (GIS), or cellular network.

This work again uses MPC for informative path planning. To ensure safety of the robot, the barrier function approach is used. The generated trajectory maximizes the probability of detection while avoiding collision with surrounding obstacles. To reduce computational burden, we first approximate the uncertainty of the target by a Gaussian Mixture Model (GMM). We then derive an analytical form of the objective function from GMM for path optimization.

4.3.1 Probabilistic Filtering Using Particle Filter

Since the robot uses the binary sensor, which has a nonlinear sensor measurement model, we use the particle filter for estimating the target position. The details of PF are presented in Section 2.2. Here we only recap the parts that are relevant to this work. Consider a particle set $\chi_k = [x_k^{g,1}, \ldots, x_k^{g,N}]$ at time k drawn from the prior distribution $P(x^g|z_{1:k})$[3]. The prior distribution can then be approximated by these particles:

$$P(x^g|z_{1:k}) \approx \sum_{v=1}^{N} \delta(x^g - x_k^{g,v}), \tag{4.6}$$

[3]note that since the target is static, x^g does not have the time index k here.

where δ represents the Dirac delta function. The particle weights for next-step resampling is computed by using the sensor measurement z_{k+1}:

$$w_{k+1}^{v} = \frac{P(z_{k+1}|x_k^{g,v}, x_{k+1}^{r})}{\sum_{v=1}^{N} P(z_k|x_{k+1}^{g,v}, x_{k+1}^{r})}.$$ [4]

A new particle set $\chi_{k+1} = [x_{k+1}^{g,1}, \ldots, x_{k+1}^{g,N}]$ at time $k+1$ is generated by drawing particles in χ_k according to weights w_{k+1}^{v} $(v = 1, \ldots, N)$. The next step probability map $P(x^g|z_{1:k+1})$ can be computed by using Eq. (4.6). However, particle representation of $P(x^g|z_{1:k})$ is inconvenient for optimization-based path planning approaches since it is computationally heavy to consider all particles, the number of which can be on the magnitude of 10^3 or even higher. So instead of using particles in the planning process, this work approximates the distribution represented by particles with a Gaussian Mixture Model (GMM), as shown in Eq. (4.7). The GMM is actually a weighted sum of multiple Gaussian distributions, of which the weights and Gaussian parameters can be computed by the Expectation-Maximization algorithm [15].

$$P(x^g|z_{1:k}) \approx \sum_{j=1}^{n} v_{j,k} q(x^g; \Phi_{j,k}), \tag{4.7}$$

where n represents the number of distributions, $v_{j,k}$ is the weight, and $\Phi_{j,k}$ stands for the parameter of Gaussian distributions, which will be described in Section 4.3.3.

4.3.2 Path Planning Using Model Predictive Control

This work considers a simple unicycle model of the robot (Eq. (2.13)), with states consisting of the xy-coordinate and the heading angle. Direct control on the speed and angular velocity is assumed. The MPC for generating the safe trajectory is shown in Eq. (4.8). The main goal of this formulation is to maximize the probability of detecting the target in the prediction horizon while avoiding collision with surrounding obstacles.

$$\min_{u_{k:k+N-1}^{r}} \; w_1 J_{\text{pnd}}(x_{k:k+N|k}^{r}, u_{k:k+N-1}^{r}) + w_2 J_{\text{col}}(x_{k:k+N|k}^{r}, u_{k:k+N-1}^{r}) + w_3 J_{\text{trm}}(x_{k:k+N|k}^{r}, u_{k:k+N-1}^{r}) \tag{4.8a}$$

$$s.t.\ x_{k+i+1|k}^{r} = f^r(x_{k+i|k}^{r}, u_{k+i}^{r}), \tag{4.8b}$$

$$x_{k+i+1|k}^{r} \in \mathcal{X},\ u_{k+i}^{r} \in \mathcal{U}, \tag{4.8c}$$

$$i = 0, \ldots, N-1,$$

where Eq. (4.8b) uses the model defined in Eq. (2.13); $\mathcal{X}$ and $\mathcal{U}$ represent the feasible sets of robot state and control input, consisting of constraints on robot position, speed, and steering rate.

The objective function Eq. (4.8a) is a weighted sum of three functions: J_{pnd}, J_{col} and J_{trm}, namely, non-detection function, collision avoidance function and terminal cost function, respectively. The first function plays a major role in target detection while the second one enforces the

[4] In this entity $x_k^{g,v}$ is used because the target is static. If the target is moving, each particle state needs to be first forward predicted, then the new particle states are used to compute the weights, as shown in Algorithm 1.

robot to avoid collision with obstacles. The third function is used to guide the robot towards high probability density region, where the target is more likely to be located. Note that the calculation of the first and third functions depends on the target probability distribution of target state, which is updated at each control step of MPC.

(1) The non-detection function is defined as

$$J_{\text{pnd}}(x^r_{k:k+N|k}, u^r_{k:k+N-1}) = P(z_{k+1:k+N} = \mathbf{0} | x^r_{k+1:k+N|k}), \tag{4.9}$$

which actually represents the probability that no target detection happens in the prediction horizon. Traditionally, the calculation of Eq. (4.9) incurs high computation burden because no analytical expression exists and numerical integration over large state space is required. This work derives an analytical form for this function by approximating the updated probability map as a Gaussian Mixture Model, which enables the use of non-myopic prediction horizon while maintaining the problem computationally tractable. This will be described in Section 4.3.3.

(2) Collision avoidance is enforced by using barrier functions in the objective function, which is defined as

$$J_{\text{col}}(x^r_{k:k+N|k}, u^r_{k:k+N-1}) = -\sum_{i=1}^{N}(\sum_{l=1}^{n_p} log(d_i^{Rp_l} - r^{Rp}) + \sum_{m=1}^{n_b} log(d_i^{Rb_m} - r^{Rb})), \tag{4.10}$$

where n_p and n_b denote the number of moving and static obstacles, respectively; r^{Rp} and r^{Rb} represent the safe distances; $d_i^{Rp_l}$ $(l = 1, \ldots, n_p)$ and $d_i^{Rb_m}$ $(m = 1, \ldots, n_b)$ stand for Euclidean distances between obstacles and the robot, defined as

$$d_i^{Rp_l} = ||x^r_{k+i|k} - x^{p_l}_{k+i|k}||_2,$$
$$d_i^{Rb_m} = ||x^r_{k+i|k} - x^{b_m}||_2,$$

where x^{b_m} represents the position of the m^{th} static obstacle and $x^{p_l}_{k+i|k}$ denotes the predicted position of the l^{th} moving obstacle in the prediction horizon. When the distance between the robot and an obstacle drops below the safe distance, Eq. (4.10) will go to infinity, which essentially prohibits the robot from colliding with any obstacle. Here a constant-velocity motion model is used to predict the future trajectory of moving obstacles, such as pedestrians and motor vehicles:

$$v_k^{p_l} = \frac{x_k^{p_l} - x_{k-1}^{p_l}}{\Delta t},\ l = 1, \ldots, n_p, \tag{4.11a}$$
$$x^{p_l}_{k+i|k} = x_k^{p_l} + i v_k^{p_l} \Delta t,\ i = 1, \ldots, N, \tag{4.11b}$$

where $v_k^{p_l}$ is the estimated velocity and $x^{p_l}_{k+i|k}$ is the predicted position at time $k+i$. For more complex prediction methods, readers can refer to [50, 22, 104].

(3) The terminal cost function $J_{\text{trm}}(x^r_{k:k+N|k}, u^r_{k:k+N-1})$ is defined as

$$J_{\text{trm}}(x^r_{k:k+N|k}, u^r_{k:k+N-1}) = ||x^r_{k+N|k} - x_k^{g,M}||_2, \tag{4.12}$$

where

$$x_k^{g,M} = \arg\max_{x^g} P(x^g|z_{1:k})$$

is the maximum a-posteriori (MAP) estimate of the probability distribution at the current step k. The terminal cost function penalizes the distance between the robot and the probability peak. Moreover, the weight w_3 can be adjusted based on the value of $J_{\text{pnd}}(x^r_{k:k+N|k}, u^r_{k:k+N-1})$, i.e., $w_3 = 0$ when $J_{\text{pnd}}(x^r_{k:k+N|k}, u^r_{k:k+N-1}) > \epsilon$, where ϵ is a user-specified threshold; otherwise, $w_3 \neq 0$. This design is able to drive the robot out of local low probability density areas towards the position with high probability density.

4.3.3 Analytical Form of Objective Function

In this section, we use the notation of the Exponential Family of Distributions[5] to derive an analytical form of the non-detection cost function (Eq. (4.9)). For the sake of simplicity, a d-dimensional Gaussian probability density function (Eq. (4.13)) with mean μ and covariance Σ is written in the form of the Exponential Family of Distributions (Eq. (4.14)):

$$q(x;\mu,\Sigma) = (2\pi)^{-\frac{d}{2}}|\Sigma|^{-\frac{1}{2}}e^{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)}, \tag{4.13}$$

$$q(x;\Phi) = e^{\Phi^T T(x) - A(\Phi)}, \tag{4.14}$$

where

$$\begin{aligned}
\Phi &= [\Lambda, \Psi]\,, \\
\Lambda &= \Sigma^{-1}\mu, \\
\Psi &= \frac{1}{2}\Sigma^{-1}, \\
T(x) &= \left[x, -xx^T\right]^T, \\
A(\Phi) &= \frac{1}{4}\Lambda^T\Psi^{-1}\Lambda + \frac{d}{2}ln(2\pi) - \frac{1}{2}|2\Psi|.
\end{aligned}$$

The GMM representation enables an analytical expression of the objective function. To be specific, Eq. (4.9) is expanded as a product of the target probability density function and the sensor model:

$$J_{\text{pnd}}(x^r_{k:k+N|k}, u^r_{k:k+N-1}) = P(z_{k+1:k+N} = \mathbf{0}|x^r_{k+1:k+N|k}), \tag{4.16a}$$

$$= \int_S P(z_{k+1:k+N} = \mathbf{0}, x^g|x^r_{k+1:k+N|k})\mathrm{d}x^g, \tag{4.16b}$$

$$= \int_S P(x^g)P(z_{k+1:k+N} = \mathbf{0}|x^g, x^r_{k+1:k+N})\mathrm{d}x^g, \tag{4.16c}$$

$$= \int_S P(x^g)\prod_{i=1}^{N} P(z_{k+i} = \mathbf{0}|x^g, x^r_{k+i})\mathrm{d}x^g. \tag{4.16d}$$

[5]Please refer to Appendix A.1 for details of the Exponential Family of Distributions.

Here, Eq. (4.16b) is obtained by using the law of total probability. Eq. (4.16c) and (4.16d) are based on the conditional independence that sensor measurement only depends on the robot and target position. By substituting approximated target PDF, Eq. (4.7), into $P(x^g)$ and using the closure of the product of Gaussian functions, an analytical form of $J_{\text{pnd}}(x^r_{k:k+N|k}, u^r_{k:k+N-1})$ can be derived:

$$J_{pnd}(x^r_{k:k+N|k}, u^r_{k:k+N-1}) = \int_S \sum_{j=1}^{n} v_{j,k} e^{\Phi_{j,k}{}^T T(x^g) - A(\Phi_{j,k})} \prod_{i=1}^{H} \left[1 - c e^{\Phi^R_{k+i}{}^T T(x^g) - A(\Phi^R_{k+i})}\right] \mathrm{d}x^g, \tag{4.17a}$$

$$\begin{aligned} &= \sum_{j=1}^{n} v_j \Big[1 - \sum_{i=1}^{H} c e^{\alpha_{ji}} + \sum_{\substack{i_1,i_2 \in \{1,\dots,H\} \\ i_1 \neq i_2}} c^2 e^{\alpha_{j,i_1,i_2}} + \dots \\ &\quad + (-1)^p \sum_{\substack{i_1,i_2,\dots,i_p \in \{1,\dots,H\} \\ i_m \neq i_n,\ m,n \in \{1,\dots,p\}, m \neq n}} c^p e^{\alpha_{j,i_1,i_2,\dots,i_p}} + \dots \\ &\quad + (-1)^H \sum_{\substack{i_1,i_2,\dots,i_H \in \{1,\dots,H\} \\ i_m \neq i_n,\ m,n \in \{1,\dots,H\}, m \neq n}} c^H e^{\alpha_{j,i_1,i_2,\dots,i_H}} \Big], \end{aligned} \tag{4.17b}$$

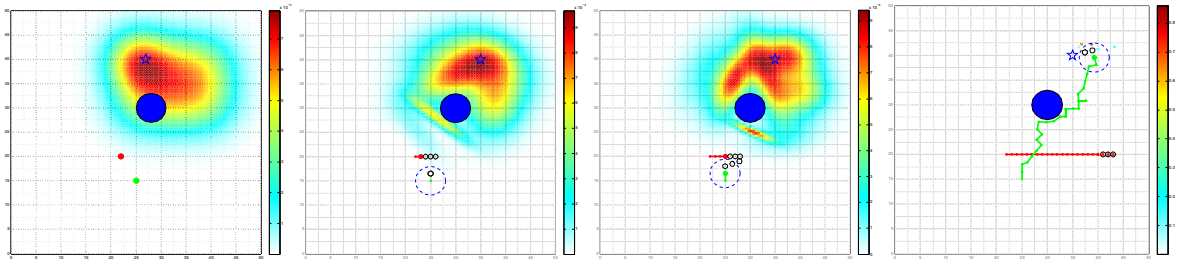
where Φ^R_{k+i} represents the parameter of binary sensor model Eq. (2.5) using the definition of Exponential Family of Distributions and $\alpha_{j,i_1,i_2,\dots,i_l}$ is defined as

$$\begin{aligned} \alpha_{j,i_1,i_2,\dots,i_l} &= A(\Phi_{sum}) - A(\Phi_{j,k}) - \sum_{s=1}^{l} A(\Phi^R_{k+i_s}), \\ \Phi_{sum} &= \Phi_{j,k} + \sum_{s=1}^{l} \Phi^R_{k+i_s}, \\ l &= 1, \dots, H. \end{aligned}$$

This analytical form Eq. (4.17b) can reduce the computation complexity for multiple-step planning because the numerical integration Eq. (4.16d) is no longer needed. The full derivation of Eq. (4.17b) is presented in the Appendix A.2.

4.3.4 SIMULATION

Two simulations have been carried out to illustrate the effectiveness of the proposed search method. The initial layout of the first scenario is shown in Fig. 4.8a. The approximated target PDF using a GMM is represented by the colored field, with the red for high probability density and the white for low probability density. The robot, shown as the green dot, starts from the lower-half of the field. A static obstacle, drawn as the large blue circle, and a moving obstacle, represented by the small red circle, exist in the field. The moving obstacle follows the constant-velocity motion model. As the search starts, the robot predicts the moving obstacle's motion with a prediction horizon of three (represented as red dots surrounded by black circles) and plans its own path (denoted



(a) (b) (c) (d)

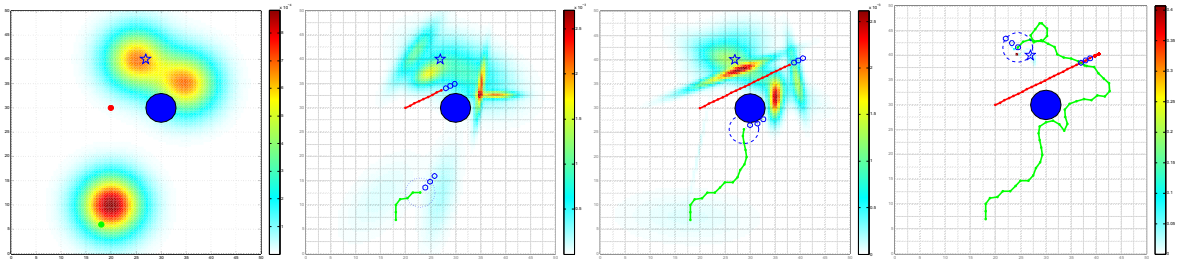
Figure 4.8: (a) The initial layout of the scenario with one high probability density region. (b) The robot starts searching; the red dots with black circles are the predicted trajectory of the moving obstacle and the green ones are the robot's planned trajectory. (c) The robot avoids collision with the human in the prediction horizon. (d) The robot successfully localizes the target.

as green dots surrounded by black circles) using the MPC, which is shown in Fig. 4.8b. The radius of black circles represents the size of the robot and the moving obstacle; a collision will occur if the black circles intersect in the prediction horizon. The large dashed blue circle around the robot represents the range within which the sensor has a high probability of target detection when the target falls inside this circle. Based on sensor measurements, the target PDF is updated using the particle filter. Collision avoidance is successfully enforced during the robot's search process, which can be verified in Fig. 4.8c that the robot detours to avoid collision with the moving obstacle in the prediction horizon. The search ends when the l_2-norm of the covariance matrix of the probability map is smaller than a threshold, which is shown in Fig. 4.8d that the robot has successfully localized the target.

The initial layout of the second scenario is shown in Fig. 4.9a, which contains two separate high probability density regions. The robot starts searching within the lower-left high probability density region, as shown in Fig. 4.9b. It follows the path that maximizes the probability of detecting the target. As the robot does not detect the target, the probability density in this region decreases and the robot will leave this region when the probability density drops below a given threshold. In Fig. 4.9c, the robot moves towards the top-right high probability region. It is worth noting that the robot's planned path is along the boundary of the static obstacle, which shows that the collision avoidance is effectively enforced. Fig. 4.9d shows that the robot has successfully localized the target.

4.4 Summary

In this chapter, we propose model predictive control (MPC)-based path planning approaches for a mobile robot to autonomously search for and localize/track a static/moving target. We first consider the case of linear sensor model and linear target motion model. We take into account the effects of sensor's limited sensing domain in the predictive horizon by utilizing the modified Kalman filter for handling intermittent measurements. To deal with the discontinuity caused by limited sensing domain, we propose a bell-shaped function to approximate the sensing domain's boundary. A two-



(a) (b) (c) (d)

Figure 4.9: (a) The initial layout of the scenario with two high probability density regions. (b) The robot searches in the lower-left region, following the path that maximizes the probability of detecting the target. (c) The robot avoids collision with the static obstacle in the prediction horizon. (d) The robot successfully localizes the target.

step sequential planning approach is then developed for solving the MPC problem. Simulation results have demonstrated the effectiveness of the proposed path planning algorithm in searching for and tracking the target.

We then subsequently consider using a mobile robot to localize a stationary target in a dynamic environment. The MPC-based planning framework is utilized to maximize the probability of detecting the target while avoiding collision with surrounding obstacles. The probability density function of target position is updated using particle filter and then approximated by a Gaussian Mixture Model (GMM) in planning process. An analytical form of the objective function in the prediction horizon is derived for the purpose of reducing the computational complexity to allow a longer prediction horizon. Simulations have shown the effectiveness of the proposed method for a robot to autonomously search for the target while avoiding collision with both static and moving obstacles.

Chapter 5

Path Planning for Human Following

5.1 Introduction

This chapter considers the path planning of human-companion robots under the uncertainty of human trajectory. In recent years, robots that can act as human companions, such as service robots, mobile teleconference robots, and elderly-care robots, have received considerable attention [35, 142, 51]. The work in this chapter aims to develop *socially desirable* companion robots, which behave in a safe and comfortable manner when accompanying a person [96, 142]. Safety concern requires that the robot avoid collisions that could injure humans or the robot itself [154]. Comfort requires robots to pose little annoyance and stress for the human [96], which is mainly related to the *human spatial behavior* [2] and is usually formulated in terms of the distance between the robot and human. It has also been considered as a contributing factor to comfort if the robot can maintain a similar speed as the human [75].

To achieve the socially desirable following behavior, the robot needs to predict human future trajectories based on the measured human's motion states, and then plan its own trajectory accordingly. The work in this chapter proposes a model-based prediction approach, namely the Parallel Interacting Multiple Model-Unscented Kalman Filter (PIMM-UKF), for human motion estimation and prediction, taking advantage of the fact that human motion usually consists of different simple motion patterns [1]. Utilizing the predicted human trajectory, a nonlinear model predictive control [60] (MPC)-based motion planner is developed which simultaneously considers the safety and comfort requirements.

The remainder of this chapter is organized as follows: first, the problem of path planning for a human-companion robot is formulated in Section 5.2. Then, the overall architecture of the path planning system, including the PIMM-UKF-based human motion predictor and MPC-based path planner, is described in Section 5.3. The simulation setup and results on evaluating the proposed approach are presented in Section 5.4.

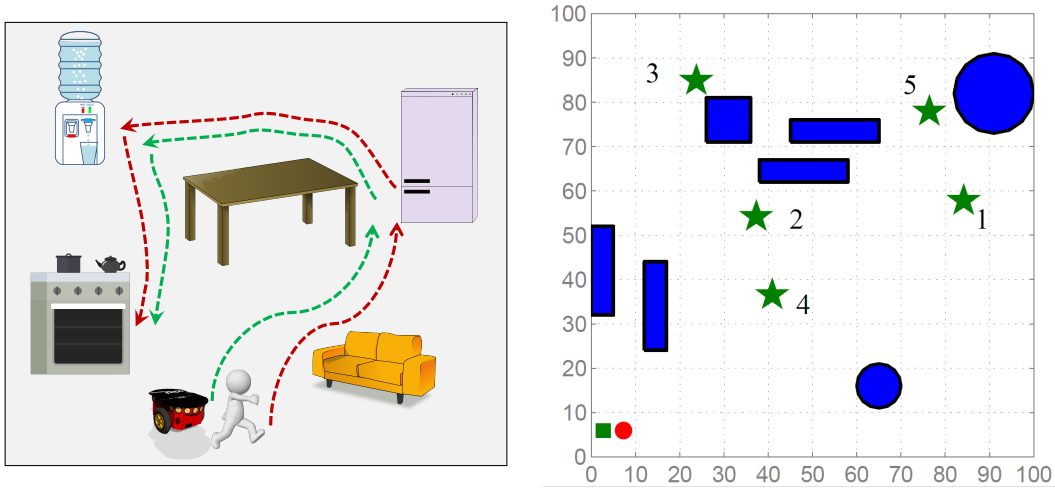


Figure 5.1: (Left) An example scenario for the companion robot to accompany a human. The person will move to several furnitures and the robot needs to accompany the person while avoiding injuring her and hitting any furnitures. (Right) An abstract representation of the scenario. The red circle and the green square represent the human and the robot companion, respectively. There are five randomly generated target positions (green stars) that the human should arrive sequentially (in the ascending order of the numbers next to each target position). Blue shapes represent obstacles.

5.2 Problem Formulation

Consider an example of cluttered 2-D space (Figure 5.1), in which a person needs to move to several destinations. A companion robot is required to accompany the human in a socially desirable way and sequentially move to these destinations. The robot has no prior knowledge about the human destinations. However, it can measure human positions in real time from the measurement tools such as cameras. Neither the robot nor human can enter or traverse obstacles, represented by blue polygons and circles. The position and dimension of obstacles are known a-priori.

When accompanying the human, the robot is expected to obey the aforementioned safety and comfort requirements, as illustrated in Figure 5.2. To be specific, a circular "unsafety zone" (blue circle) with radius d_s is defined around the target person. Stepping into the "unsafety zone" is considered as risky behavior, and thus should be strictly prohibited. The comfort requires the robot to stay within a "comfort zone" (orange area) around the human, and also keep similar speed. This work adopts the "Proxemics" model, proposed by Hall et al. [69], for designing the "comfort zone" in the way similar to works on human-robot interaction [123, 51]. Comfort is a soft requirement in that a robot may move outside of the "comfort zone" (Figure 5.2(c)), though not desirable.

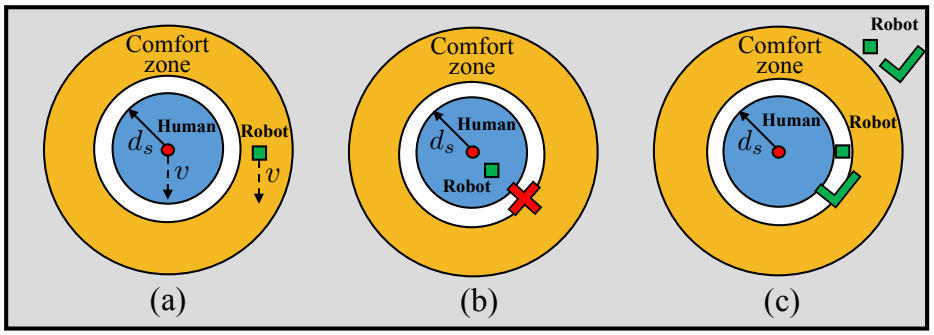


Figure 5.2: Illustration of safety and comfort requirements. (a) The most desirable companion behavior is that the robot stays within the "comfort zone" and maintains similar speed as the human. (b) The robot is not allowed to enter the "unsafety zone". (c) It is acceptable that the robot stays outside of the "comfort zone" and the "unsafety zone".

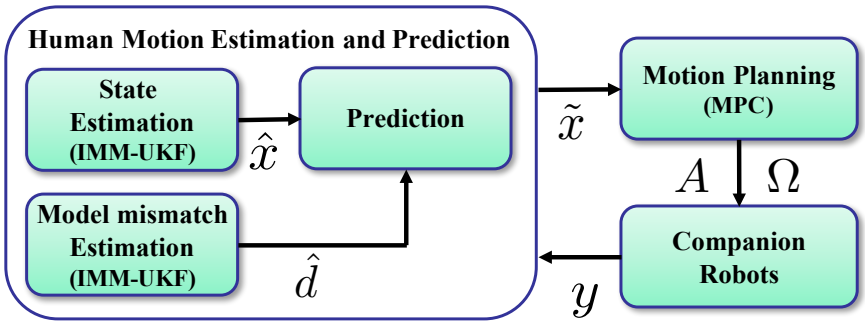


Figure 5.3: Diagram of the overall system of the companion robot.

5.3 Path Planning System[1]

5.3.1 System Overview

The architecture of the proposed path planning system is shown in Figure 5.3. The system consists of two intertwined components: the module for human motion estimation and prediction and the module for robot motion planning. For the former, we propose a parallel-estimation structure, denoted as the "State Estimation" block and the "Model mismatch Estimation" block in Figure 5.3, for estimating human motion states and the model mismatch, respectively. These estimates are then used for probabilistic prediction of human trajectory in the near future. The "Motion Planning" block utilizes the forecasted evolution of human motion states to design the robot companion behavior via optimizing the acceleration and the orientation of the robot under the MPC framework. We assume that the robot can measure the longitudinal and lateral position of the human using cameras.

[1]This section is based on the work published in [104].

5.3.2 Human Motion Estimation

We extend the Interacting Multiple Model (IMM) approach for human motion estimation. In this work, IMM incorporates two typical motion models and dynamically adjusts the mode probabilities based on the observed human trajectory [178]. To deal with the nonlinearity of the human motion, such as making turns, an Unscented Kalman Filter (UKF) is applied to each model in the IMM framework, resulting in the so-called IMM-UKF structure. In order to account for the effects of unmodeled dynamics of a human, we design a model mismatch estimator that augments the IMM-UKF structure, which further improves the prediction of human motion. Two independent IMM-UKF estimators are thus used in tandem to estimate states and model mismatch simultaneously, and then systematically predict the future motion of the human, which forms the Parallel IMM-UKF (PIMM-UKF) approach.

State Estimation

The IMM contains two different kinematic models: one for the coordinated turn motion model, reflecting the action of making turns or moving along a curved path, and the other for the uniform motion model, representing the constant heading movement.

The equation for the coordinated turn motion model is shown below:

$$x^{h,1}_{s,k+1} = f^{h,1}_s(x^{h,1}_{s,k}) + G_s w_{1,k} \tag{5.1a}$$

$$f^{h,1}_s(x^{h,1}_{s,k}) = \begin{bmatrix} p^h_1 + \frac{\sin(\omega^h T)}{\omega^h}\mu^h_1 - \frac{1-\cos(\omega^h T)}{\omega^h}\mu^h_2 \\ \cos(\omega^h T)\mu^h_1 - \sin(\omega^h T)\mu^h_2 \\ p^h_2 + \frac{1-\cos(\omega^h T)}{\omega^h}\mu^h_1 + \frac{\sin(\omega^h T)}{\omega^h}\mu^h_2 \\ \sin(\omega^h T)\mu^h_1 + \cos(\omega^h T)\mu^h_2 \\ \omega^h \end{bmatrix} \tag{5.1b}$$

$$G_s = \begin{bmatrix} \frac{T^2}{2} & 0 & 0 \\ T & 0 & 0 \\ 0 & \frac{T^2}{2} & 0 \\ 0 & T & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.1c}$$

$$w_1 \sim \mathcal{N}(0, Q^1_s),$$

and the equation of the uniform motion model is represented as follows:

$$x^{h,2}_{s,k+1} = f^{h,2}_s(x^{h,2}_{s,k}) + G_s w_{2,k} \tag{5.2a}$$

$$f^{h,2}_s(x^{h,2}_{s,k}) = \begin{bmatrix} p^h_1 + \mu^h_1 T \\ \mu^h_1 \\ p^h_2 + \mu^h_2 T \\ \mu^h_2 \\ 0 \end{bmatrix} \tag{5.2b}$$

$$w_2 \sim \mathcal{N}(0, Q^2_s),$$

where $x_{s,k}^{h,i}$, $i = 1, 2$ represents the human motion state including five elements: $\left[p_1^h, \mu_1^h, p_2^h, \mu_2^h, \omega^h\right]^T$ [2], where p_1^h, p_2^h denote the longitudinal and lateral position, μ_1^h, μ_2^h the corresponding velocity and ω^h the turn rate; T is the sampling time; $w_{i,k}$, $i = 1, 2$ represents process noise that follows a zero-mean Gaussian distribution with Q_s^i being the covariance matrix.

The uniform motion model is essentially a special case of the coordinated turn motion model with the turn rate ω being fixed to zero. It seems that only considering the coordinated turn motion model suffices to estimate human motion states, in addition to the benefits of reduced computations by using the single model. However, including two models are necessary since this allows the estimator for fast detection of change of motions, which is shown in [26].

The sensor observation model is represented as :

$$z_{s,k}^{h,i} = C_s x_{s,k}^{h,i} + v_k,$$

where $z_{s,k}^{h,i}$ denotes the observed human state at time step k; v_k stands for measurement noise. By using camera sensors, the human positions can be directly measured. Therefore, the parameters in the observation model is defined as:

$$C_s = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}, v_k \sim \mathcal{N}(0, V_s),$$

where V_s is the convariance matrix of the measurement noise. Note that both $z_{s,k}^{h,1}$ and $z_{s,k}^{h,2}$ are always equivalent, coming from the actual measurement.

Model mismatch estimation

A model mismatch estimation algorithm that is able to capture unmodeled dynamics of humans is proposed for the purpose of improving human motion prediction. The augmented state formulation is one of the most popular approaches used to compensate for model plant mismatch[169, 172, 85]. The model mismatch estimator is obtained by appending each motion model in IMM-UKF with three additional states, corresponding to the mismatch of longitudinal position, lateral position and turn rate. By applying IMM-UKF to the augmented system, each additional state can be estimated online, and then added to the predicted human positions, velocities and turn rates as corrective terms.

At the first sight, it seems that merely using the augmented IMM-UKF estimator is sufficient for estimating both states and model mismatch. However, it is necessary to maintain two separate estimators that correspond to the state estimator and the model mismatch estimator, which stems from the trade-off between two main concerns in designing estimators: the fast response and the noise rejection. In this work, the model mismatch estimator needs to track human trajectories as fast as possible to precisely measure model mismatch so as to compensate for the inaccurate prediction of the human motion. However, the requirement of fast tracking usually results in very noisy human state estimation, which is not suitable for reliable prediction. Thus, we propose the Parallel IMM-UKF framework to simultaneously produce estimates of states and model mismatch.

[2] We omit k and i in the notation for the purpose of notational simplicity.

The state estimation IMM-UKF is designed to reject the noise from observations with slower tracking response than the model mismatch estimation. The model mismatch estimation IMM-UKF, on the other hand, is designed to achieve fast tracking of human motion states even if the state estimation results are noisy. This is achieved by setting a large process noise for the additional three states while keeping the process noise for the original states the same as the ones used in the state estimator. The augmented coordinated turn motion for the model mismatch estimation is shown below:

$$
\begin{aligned}
x^{h,1}_{d,k+1} &= f^{h,1}_d(x^{h,1}_{d,k}) + G_d w_{1,k} \\
f^{h,1}_d(x^{h,1}_{d,k}) &= \begin{bmatrix}
p^h_1 + \frac{\sin(\omega^h T)}{\omega^h}\mu^h_1 - \frac{1-\cos(\omega^h T)}{\omega^h}\mu^h_2 + \frac{T^2}{2}d^h_1 \\
\cos(\omega^h T)\mu^h_1 - \sin(\omega^h T)\mu^h_2 + T d^h_1 \\
p^h_2 + \frac{1-\cos(\omega^h T)}{\omega^h}\mu^h_1 + \frac{\sin(\omega^h T)}{\omega^h}\mu^h_2 + \frac{T^2}{2}d^h_2 \\
\sin(\omega^h T)\mu^h_1 + \cos(\omega^h T)\mu^h_2 + T d^h_2 \\
\omega^h + T d^h_3 \\
d^h_1 \\
d^h_2 \\
d^h_3
\end{bmatrix} \\
G_d &= \begin{bmatrix}
\frac{T^2}{2} & 0 & 0 & 0 & 0 & 0 \\
T & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{T^2}{2} & 0 & 0 & 0 & 0 \\
0 & T & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & T & 0 & 0 \\
0 & 0 & 0 & 0 & T & 0 \\
0 & 0 & 0 & 0 & 0 & T
\end{bmatrix} \\
w_1 &\sim \mathcal{N}(0, Q^1_d).
\end{aligned}
$$

The augmented uniform motion for model mismatch estimation is

$$
\begin{aligned}
x^{h,2}_{d,k+1} &= f^{h,2}_d x^{h,2}_{d,k} + G_d w_{2,k} \\
f^{h,2}_d(x^{h,2}_{s,k}) &= \begin{bmatrix}
p^h_1 + \mu^h_1 T + \frac{T^2}{2}d^h_1 \\
v^h_1 + T d^h_1 \\
p^h_2 + \mu^h_2 T + \frac{T^2}{2}d^h_2 \\
v^h_2 + T d^h_2 \\
\omega^h + T d^h_3 \\
d^h_1 \\
d^h_2 \\
d^h_3
\end{bmatrix} \\
w_2 &\sim \mathcal{N}(0, Q^2_d),
\end{aligned}
$$

where $x^{h,i}_{d,k}$ $(i = 1, 2)$ represents the human motion state including eight elements in the model mismatch estimation: $\left[p^h_1, \mu^h_1, p^h_2, \mu^h_2, \omega^h, d^h_1, d^h_2, d^h_3\right]^T$, where the first five states are defined as before,

and d_1^h, d_2^h, d_3^h represent the model mismatch of longitudinal, lateral and angular components, respectively; $w_{i,k}\,(i = 1, 2)$ represents process noise that follows a zero-mean Gaussian distribution with Q_d^i being the covariance matrix.

The observation model for the model mismatch estimation is represented as

$$z_{d,k}^h = C_d x_{d,k}^h + v_k,$$

where $z_{d,k}^h$ denotes the observed human state at the time step k. The parameters in the observation model is defined as:

$$C_d = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, v_k \sim \mathcal{N}(0, V_d),$$

where V_d is the convariance matrix of the measurement noise.

5.3.3 Human Motion Prediction

The PIMM-UKF is then used for predicting human motion, incorporating the estimated human motion states and model mismatch. To be specific, using the uniform motion model and the turn motion model, human positions and velocities in each model can be extrapolated based on the current estimated state and model mismatch. Prediction from each model are then combined according to the mode probabilities to obtain the overall predicted human motion state. Let $\hat{x}_{s,k|k}^{h,j}$, $\hat{x}_{s,k+i|k}^{h,j}$ and $\hat{d}_{k|k}^{h,j}$ represent the estimated and predicted human states and the estimated model mismatch associated with the j^{th} model at time k and $k + i\,(i \geq 0)$, respectively, based on the observations up to time k. The prediction procedure works as follows:

$$\hat{x}_{s,k+l+1|k}^h = \sum_{j=1}^{r} \mu_j \hat{x}_{s,k+l+1|k}^{h,j}, \tag{5.4a}$$

$$\hat{x}_{s,k+l+1|k}^{h,j} = f_s^j(\hat{x}_{s,k+l|k}^{h,j}) + h_s^j(\hat{d}_{k|k}^{h,j}), \quad l = 0, \ldots, N-1 \tag{5.4b}$$

$$h_s^j(\hat{d}_{k|k}^{h,j}) = \begin{bmatrix} \frac{T^2}{2}\hat{d}_1^{h,j} \\ T\hat{d}_1^{h,j} \\ \frac{T^2}{2}\hat{d}_2^{h,j} \\ T\hat{d}_2^{h,j} \\ T\hat{d}_3^{h,j} \end{bmatrix},$$

where N denotes the prediction horizon; r is the number of models. f_s^j represents each model in Eq. (5.1) and Eq. (5.2).

5.3.4 Path Planning for Human Following

To design the MPC controller, we first obtain the estimated and predicted human states, $\hat{x}_{s,k+i|k}^h$ for horizon $i = 0 \ldots N$ at time step k from PIMM-UKF according to Eq. (5.4). The state contains

five elements, which are $\left[\hat{p}_1^h,\ \hat{\mu}_1^h,\ \hat{p}_2^h,\ \hat{\mu}_2^h,\ \hat{\omega}^h\right]^T$. We calculate the tracking reference of the human position vector and velocity as follows:

$$\hat{p}^h_{k+i|k} = \begin{bmatrix} \hat{p}^h_{1,k+i|k} \\ \hat{p}^h_{2,k+i|k} \end{bmatrix}, \quad i = 0 \dots N$$
$$\hat{\mu}^h_{k+i|k} = \left\| \begin{bmatrix} \hat{\mu}^h_{1,k+i|k} \\ \hat{\mu}^h_{2,k+i|k} \end{bmatrix} \right\|_2, \quad i = 0 \dots N$$

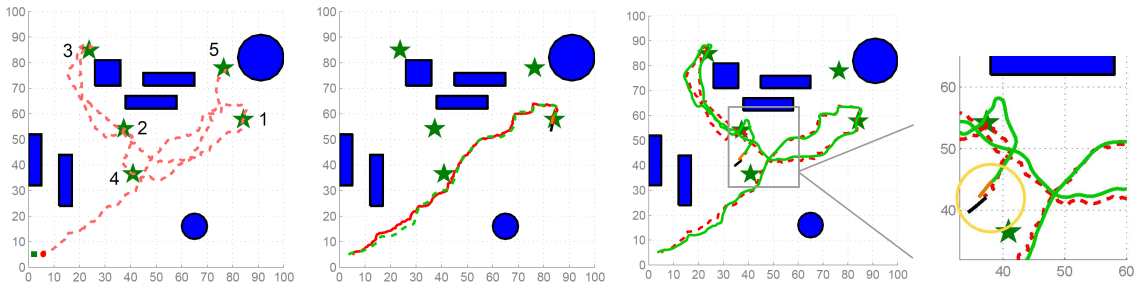
Then, we can compute the current optimal control input at time step k by solving the following FHCOCP that incorporates the kinematics of the robot and the safety and comfort requirements:

$$\min_{\mathbf{A}_k, \mathbf{\Omega}_k} \sum_{i=0}^{N} q_1 \left| \left\| p^r_{k+i|k} - \hat{p}^h_{k+i|k} \right\|_2^2 - d_c^2 \right| + q_2(\mu^r_{k+i|k} - \hat{\mu}^h_{k+i|k})^2 + q_3({a^r_{k+i|k}}^2 + {\omega^r_{k+i|k}}^2) \tag{5.5a}$$
$$\text{s.t. } x^r_{k+i+1|k} = f^r(x^r_{k+i|k}) \tag{5.5b}$$
$$a_{lb} \le a^r_{k+i|k} \le a_{ub}, \quad w_{lb} \le \omega^r_{k+i|k} \le w_{ub},\ i = 0, \dots, N-1 \tag{5.5c}$$
$$\| p^r_{k+i|k} - \hat{p}^h_{k+i|k} \|_2 > d_s,\ i = 0, \dots, N \tag{5.5d}$$
$$h_{l_s}(p^r_{k+i|k}) > 0,\ l_s = 1, \dots, n_s \tag{5.5e}$$
$$p^r_{k|k} = p^r_k, \mu^r_{k|k} = \mu^r_k, \theta^r_{k|k} = \theta^r_k, \tag{5.5f}$$

where $p^r_{k+i|k}$, $\mu^r_{k+i|k}$ and $\theta_{k+i|k}$ for $i = 1 \dots N$ represent the planned position, velocity and heading angle of the robot at time $k+i$; n_s is the number of obstacles. The obstacles are approximated by ellipses, and $h_{l_s}(\cdot)$ denotes the analytical function of the ellipse approximation of the l_s-th obstacle. The details of approximation are described in the Appendix A.3.

The objective function Eq. (5.5a) consists of three terms: the first one stands for the difference between the squared human-robot distances and the squared comfort distance d_c, which could takes any value within the "comfort zone"; the second one represents the velocity difference between the robot and the human. These reflect the comfort requirement that the robot stay within the "comfort zone" around the human, and keep similar pace at the same time. Finally, the last term penalizes the control inputs. q_1, q_2 and q_3 denote the positive weights for these three terms. The robot kinematics (Eq. (5.5b)) are defined in Eq. (2.12), with Eq. (5.5c) being the upper and lower bounds for the control inputs. The safety constraints are imposed in Eq. (5.5d) that requires the robot stay at least the safety distance d_s from the human in order to avoid collision. Eq. (5.5e) enforces the collision avoidance with obstacles by demanding that, at each prediction step, the robot should not intersect with obstacles. Eq. (5.5f) initializes the robot's planned states based on its state at time k.

The solution of the optimal control input, $(\mathbf{A}_k^*, \mathbf{\Omega}_k^*)$, consists of the sequence of optimal acceleration $a^{r*}_{k+i|k}$ and angular velocity $\omega^{r*}_{k+i|k}$ of the robot in the prediction horizon $i = 0 \dots N-1$. To reduce the formulation complexity of the FHCOCP, we first solve the problem without the obstacle constraints Eq. (5.5e). After obtaining the optimal solutions, $(\mathbf{A}_k^*, \mathbf{\Omega}_k^*)$, we check if the predicted positions violate any obstacle constraint. If violation happens, FHCOCP at time k is re-computed with the corresponding obstacle constraints being activated. When there is no any obstacle constraint violation, $a^{r*}_{k|k}$ and $\omega^{r*}_{k|k}$ are implemented as the robot's control input.



(a) (b) (c) (d)

Figure 5.4: A screenshot of the simulation. (a) The red broken line represents the human trajectory and the green solid line shows the companion robot's trajectory. (b) Both human and robot arrives at the first target. (c) Human is heading to fourth target. (d) The black line shows the predicted trajectory of the human from PIMM-UKF and the brown line shows the planned trajectory of the robot using MPC.

5.4 Simulation Results and Discussion

5.4.1 Simulation setup

The proposed motion planner for human companion robots is evaluated in 10 randomly generated scenarios, each with the size of $100m \times 100m$. Five destinations that the human needs to move to sequentially were randomly chosen. In each scenario, there were 7 obstacles with polygon or circular shapes and of different sizes. They were randomly placed in the way that no obstacle overlapped. The trajectory of the human was generated by the RRT algorithm [97], with speed randomly varying from $1m/s$ to $1.5m/s$. The RRT algorithm realistically simulated human-like motion trajectories that included straight-line motion, curved motion, change of speeds and making abrupt turns. One of the scenarios is shown in Figure 5.4a, with human trajectory represented by the red broken line. The safety distance d_s was set to be $1m$. According to the research conducted by Hall et al. [69], the comfort distance ranges from $1.2m$ to $3.6m$. In the simulation, d_c was chosen as $2.8m$, and any distance within $[1.2m, 3.6m]$ was considered a comfortable one. The GPS sensor or camera sensor for locating the human had a sampling rate of $20Hz$. The robot's maximum acceleration and deceleration were set to be $1m/s^2$ and $-3m/s^2$ respectively, and the angular velocity range was chosen to be $[-90s, 90s]$. The process noise and measurement noise for the IMM estimator were set to be 1.5×10^{-2} and 1.5, respectively. The parameters of UKF in the estimator were $L = 5, \alpha = 0.001, \kappa = 0$ and $\beta = 2$. The prediction horizon for the human motion was $2.5s$ and the robot iteratively computed the control inputs every $500ms$.

5.4.2 Simulation results

Figures 5.4a to 5.4d show both the human and robot's trajectories for one of the 10 scenarios. The performance of human motion estimation is first evaluated by comparing IMM-UKF with each single model-based UKF. The performance of PIMM-UKF is then compared with IMM-UKF for

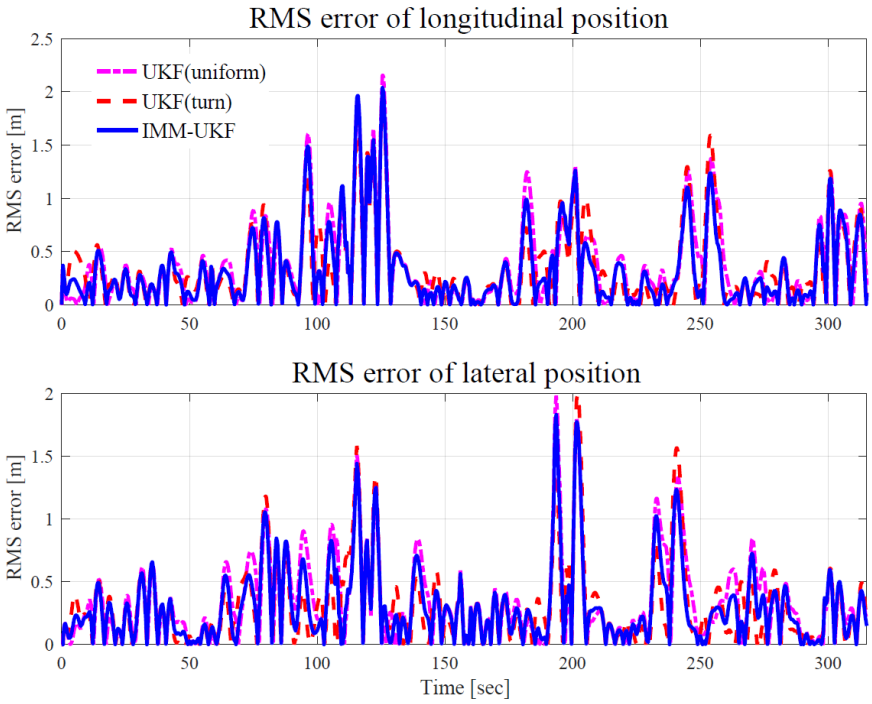


Figure 5.5: Comparison of RMS error of estimated positions between each single model-based UKF and IMM-UKF

.

Table 5.1: RMS error of the estimated position and velocity

	Direction	UKF(uniform)		UKF(turn)		IMM-UKF	
		mean	std.	mean	std.	mean	std.
Position [m]	Long.	0.52	0.04	0.48	0.02	0.46	0.03
	Lat.	0.51	0.05	0.46	0.05	0.46	0.05
Velocity [m/s]	Long.	0.59	0.03	0.60	0.02	0.58	0.02
	Lat.	0.57	0.03	0.57	0.03	0.56	0.03

human motion prediction. The MPC-based robot motion planning method is then evaluated under two different cases: one with the prediction and other without the prediction.

Human motion estimation

The error between the estimated and the actual human position and velocity at each time step is compared to evaluate the accuracy of the estimation. The position error vector is defined as:

$$\Delta_p^{est}(k) = p_k^h - \hat{p}_{k|k}^h,$$

where p_k^h denotes the actual human position at time k.

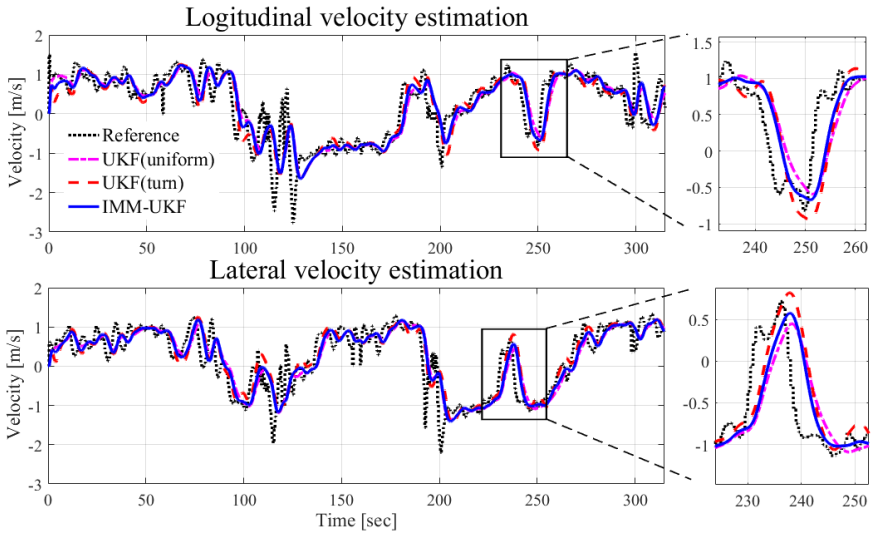


Figure 5.6: Comparison of the estimated velocity between each single model-based UKF and IMM-UKF.

Figure 5.5 shows the root mean square (RMS) errors of estimated positions on longitudinal and lateral directions using three different estimators: uniform motion model-based UKF, turn motion model-based UKF and IMM-UKF. It is easy to notice that IMM-UKF achieves smaller RMS position errors than both single model-based UKFs. IMM-UKF generates more accurate estimation when the human suddenly changes the motion, such as at time 200 and 250, in particular.

Figure 5.6 compares the velocity estimation using the three different estimators. Overall, the IMM-UKF estimator shows faster response than the uniform motion model-based UKF, though with the trade-off of small overshoot. When compared to the turn motion model-based UKF, as shown in the enlarged plots, IMM-UKF achieves smaller overshoots and faster convergence to the true velocity. This makes sense as the IMM-UKF estimator incorporates uniform motion model that can capture the sudden velocity changes.

The average and standard deviation of the RMS errors of the estimated position and velocity for the 10 randomly generated scenarios are compared in Table 5.1. IMM-UKF achieves smaller average error than each single model based-UKF, which implies that IMM-UKF in average provides improved position and velocity estimation in both directions. However, we want to point out that the improvement of IMM-UKF is not statistically significant. Further study with actual experiment data is necessary to evaluate the effectiveness of IMM-UKF.

Human motion prediction

Figure 5.7 shows the human motion prediction for the scenario in Figure 5.4a. The enlarged figure of the top-left corner is shown in Figure 5.8. Black solid line represents measurements from sensors, and pink color circles show the estimated positions. Five-step predicted positions are shown as two types of line: red broken lines represent prediction from IMM-UKF and blue solid lines stand for results from PIMM-UKF, taking model mismatch estimation into account. It

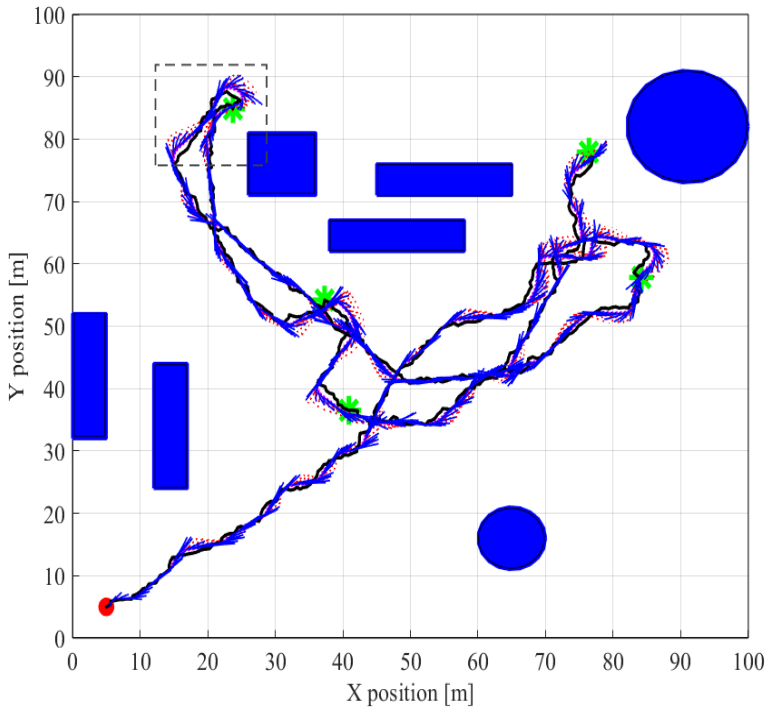


Figure 5.7: Illustration of the human motion prediction.

shows that the prediction with model mismatch estimation is closer to the actual human motion states than the other method, especially when the human motion deviates from the two considered models. To quantitatively evaluate the prediction of different approaches, we first compare the prediction errors over the prediction horizon between single model-based UKFs and IMM-UKF, then we subsequently compare the metrics between PIMM-UKF and IMM-UKF. At time k, the prediction error is defined as:

$$\Delta_{p,k}^{pre} = \frac{1}{N}\sum_{i=1}^{N} \|\tilde{p}_{k+i|k}^{h} - p_{k+i}^{h}\|_2.$$

Figure 5.9 shows the comparison of predicted position errors using each single model-based UKF, IMM-UKF and PIMM-UKF. It can be noticed that each single model-based UKF generate larger prediction error than IMM-UKF, especially when the human makes sharp turns, such as at time 120 and 240 in the top plot. Moreover, the bottom plot in Figure 5.9 shows that PIMM-UKF outperforms IMM-UKF for the most of parts. For the 10 randomly generated scenarios, Table 5.2 shows that the average errors of the position of PIMM-UKF are smaller than those of IMM-UKF, which demonstrates that the model mismatch estimation improves the human motion prediction in average, though the improvement is not statistically significant.

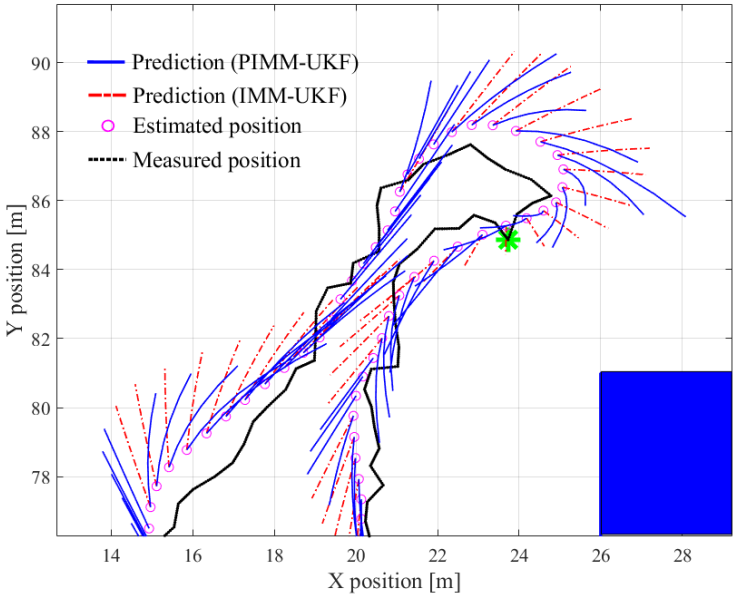


Figure 5.8: Comparison of the human motion prediction using IMM-UKF and PIMM-UKF.

Robot motion planning

Figure 5.4d shows the planned trajectory of the companion robot that accompanies the target person moving in the field. The trajectory is generated by the proposed MPC motion planner with five-step planning horizon ($2.5s$) and using the proposed PIMM-UKF approach for predicting the human trajectories. The performance of the motion planning is evaluated using the criterion of safety and comfort. To be specific, the distance and velocity differences between the robot and the human at each time step are computed using:

$$\Delta_{d,k} = \|p_k^r - p_k^h\|_2,$$
$$\Delta_{v,k} = \mu_k^r - \mu_k^h.$$

The performance is compared with non-prediction case, in which the robot does not predict the human motion and only utilizes the human's current state as a reference to make a one-step horizon ($0.5s$) motion planning.

Figure 5.10 shows the comparison results of the distance and velocity differences between the human and the robot. We can see that the MPC planner is able to ensure the safety of the accompanied human with both two strategies for the selected scenario here: no distance between the human and the robot drops below the safety distance (1 m). However, it is worth noting that the non-prediction planner actually has a higher chance to violate the safety constraint since it does not include the future prediction into the planning (See Figure 5.11 for another scenario). Additionally, by incorporating the human motion prediction via PIMM-UKF approach into the motion planning, the companion performance of the robot is significantly improved from the non-prediction case.

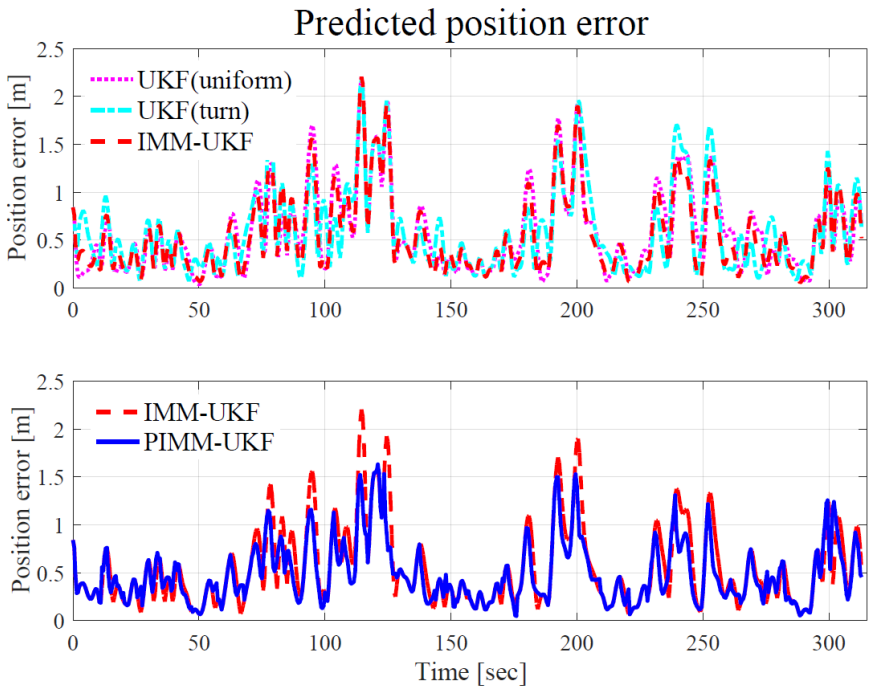


Figure 5.9: Comparison of predicted position error using each single model-based UKF, IMM-UKF and PIMM-UKF.

Table 5.2: Error of the predicted position and velocity

	Direction	IMM-UKF		PIMM-UKF	
		mean	std.	mean	std.
Position [m]	Long.	0.37	0.32	0.34	0.26
	Lat.	0.36	0.32	0.32	0.25
Velocity [m/s]	Long.	0.27	0.14	0.27	0.14
	Lat.	0.26	0.14	0.25	0.13

As we can see, the robot can stay closer to the human while maintaining a comfortable distance almost all the time. With the prediction, the robot can also respond faster when the human suddenly changes the motion pattern (i.e. the sharp turns). In fact, we can see from Figure 5.10 that the robot can catch up with the human easily compared to the non-prediction case around 200 sec. and 250 sec. at the 3^{rd} and 4^{th} targets.

Table 5.3 shows the mean and the standard deviation values of the distance and velocity differences between the human and the robot for the 10 scenarios. Overall, smaller average distance and velocity differences (2.37 m and −0.05 m/s) can be achieved using the PIMM-UKF prediction approach, compared to the values of the non-prediction case (3.35 m and −0.12 m/s).

Different from the out-performance in the mean values of the distance and velocity differences, the PIMM-UKF prediction approach achieves a larger standard deviation in velocity difference

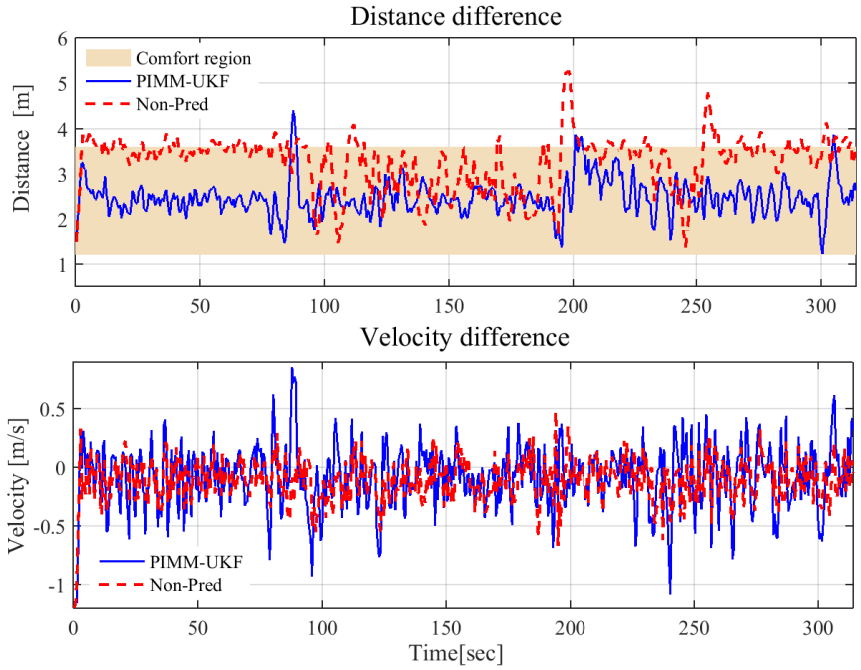


Figure 5.10: Comparison of distance and velocity differences between the human and the robot of one scenario.

Table 5.3: Performance of two MPC-based motion planners

	PIMM-UKF prediction		Non-prediction	
	mean	std.	mean	std.
Distance diff. (m)	2.37	0.49	3.35	0.54
Velocity diff. (m/s)	-0.05	0.26	-0.12	0.20

(0.26 m/s) than the one in the non-prediction case (0.20 m/s). This seems to imply that the non-prediction method is preferable for the robot in the sense of keeping stable pace as the human. However, it should be noted that such velocity variance leads to large distances between the human and the robot that are out of the "comfort zone", as shown in Figure 5.10 and Figure 5.11, which is undesirable. In fact, the change of the velocity when using the prediction results from the motion planner's effort to keep the robot within the proximity of the human. Therefore, the simulation results show the effectiveness of the MPC motion planner and the benefits of incorporating the PIMM-UKF prediction into the planning process.

5.5 Summary

This chapter has presented an autonomous path planning scheme for human-companion robots to accompany a person in a socially desirable manner. A Parallel Interacting Multiple Model-

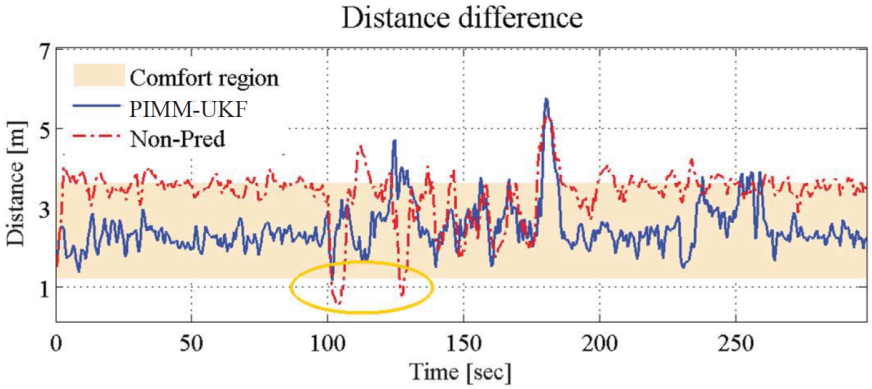


Figure 5.11: Comparison of distance difference between the human and the robot for another scenario.

Unscented Kalman Filter (PIMM-UKF) approach is proposed for human motion prediction. It combines two independent IMM-UKF estimators to estimate the human motion states and model mismatch simultaneously. Both IMM-UKF estimators incorporate the uniform motion model and the coordinated turn motion model to allow the mixed system dynamics. Moreover, a corrective term provided by the model mismatch estimator is added into the human motion prediction. Such prediction framework captures different human motion patterns and the unmodeled dynamics, thus being able to obtain more accurate prediction. Based on the predicted results, the model predictive control (MPC) is used to plan the robot's trajectory. Both the safety and comfort requirements are formulated into a finite-horizon constrained optimal control problem.

The proposed motion planning framework is evaluated using 10 randomly generated scenarios in simulation. The results show superior performance in terms of the accuracy and response time in the estimation and prediction using PIMM-UKF approach compared to IMM-UKF and single-model-based-UKF methods, especially when the human makes curved motions or sharp turns. Moreover, the MPC planner is evaluated using the PIMM-UKF and non-prediction methods. The planner successfully ensures the safety of the accompanied person using these two prediction strategies, but the PIMM-UKF results in a better robot motion behavior by keeping the robot within the comfort zone from the human. Simulation results have shown the effectiveness of the proposed scheme in generating safe and comfortable human-following behavior.

Chapter 6

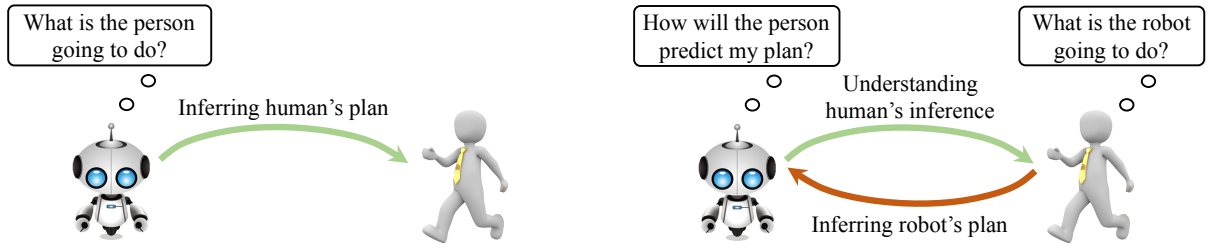
Task Planning for Human-Robot Collaboration

6.1 Introduction

With robots becoming more capable and versatile, they are now stepping into mixed workplaces shared with human beings, which has led to increasing interests in human-robot collaboration [53, 39, 150]. This chapter focuses on the task planning in human-robot collaboration, which considers the use of a human and a robot to collaboratively complete a set of given tasks. Because right now humans and robots have different capabilities, the set of tasks that each agent can complete can be different. This requires each agent to strategically choose the tasks that they are competent at carrying out to ensure that the overall performance of the team is optimized.

Collaboration relies on coordination and the ability to infer each other's intentions and adapt. In this chapter, we consider two different aspects in human-robot collaboration that takes into account the intention and inference of humans, as shown in Section 6.1. In the first aspect, we consider an assistive robot that helps a human complete a set of heterogeneous tasks. The robot constantly infers the human's goal task and adapts its own task plan to accommodate to the human's plan (Figure 6.1a). In the second aspect, we consider the inverse (Figure 6.1b). The work in this part first builds an analytic model of how a human infers a robot's plan, and then proposes a new metric, called the *t-predictability*, and a *t*-predictable planner to enable the robot to generate human-predictable task plans. The underlying goal of this work is to make robots plan in a human-predictable manner to avoid misalignment between robot's plan and human's belief. We envision that ultimately these two approaches will need to work in conjunction to achieve fluent collaboration.

Specifically, Section 6.2 describes an adaptive task planning scheme that we propose for a robot to coordinate with a person to complete a set of heterogeneous tasks. An online evaluation is provided and analyzed. Section 6.3 proposes the t-predictability and the *t*-predictable task planner to compute human-predictable plans. Both the online experiment and in-lab user study are described and analyzed.



(a) The robot infers the person's goal and then adjusts its own plan to assist the person.

(b) The robot generates a plan that is easy for the person to predict.

Figure 6.1: Two aspects in human-robot collaboration that takes human intention into account.

6.2 Adaptive Task Planning[1]

6.2.1 Introduction

This work proposes a task allocation approach that utilizes Bayesian goal inference at the motion level to enable the robot to adapt to the human at the task planning level. Across robotics[184, 45], plan recognition[29], and cognitive science [9, 136], Bayesian goal inference has been applied to the problem of inferring human goals from their ongoing actions. Previous works that use Bayesian goal inference in human-robot interaction have mainly focused on assisting humans to achieve an immediate goal [44, 49]. This work is novel in that it combines two levels of planning that are traditionally studied in isolation, namely applying goal inference at the motion level to compute a collaborative plan at the task level.

To evaluate the proposed approach, this work developed and ran a behavioral experiment (Fig. 6.2a) in which participants collaborated with three different types of virtual robots: (i) the *predictive* robot performed goal inference and re-planned as the human was moving; (ii) the *reactive* robot only re-planned after the person reached their goal; and (iii) the *fixed* robot never re-planned unless the current plan became infeasible. We also had two between-subjects conditions in which we varied the type of inference used by the robot. In the "oracle" condition, the predictive robot had direct access to the participant's intent, emulating perfect goal inference, while in the "Bayesian" condition, the predictive robot used probabilistic inference on noisy human trajectories (as would happen in a realistic setting). Building on literature that evaluates human-robot collaboration [79, 42], we evaluated performance both objectively and subjectively.

6.2.2 Task Allocation During Human-Robot Collaboration

We consider a task allocation scenario with both individual and shared tasks, in which a human agent H and a robot agent R are in a compact planar domain $\mathcal{S} \subset \mathbb{R}^2$ with N one-agent tasks located at positions $x^{o,i} \in \mathcal{S}$, $i = 1, ..., N$, and M joint tasks with locations $x^{q,j} \in \mathcal{S}$, $j = 1, ..., M$. One-agent tasks are executed instantly when visited by any agent, whereas joint tasks are only completed

[1]This section is based on the work published in [116, 109].

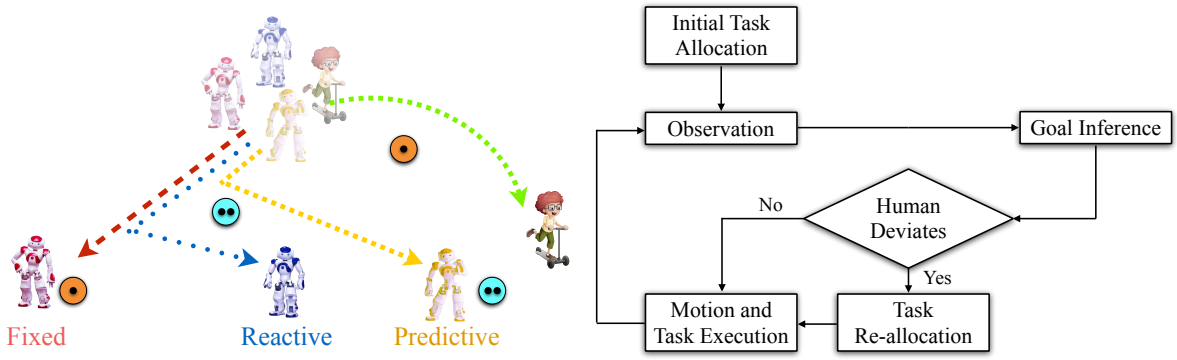


Figure 6.2: Left: a task allocation scenario that contains a human and robots. Three kinds of robot are shown here (the red, blue, and yellow robot), which are used in the online experiment. The person works with one of the robots at a time. Circles with a single dot represent one-agent tasks and circles with two dots represent joint tasks. Right: the adaptive task planning scheme.

when visited by both agents simultaneously. The agent that arrives first at a joint task cannot leave until the other one comes to help complete the task. The human starts at position $x_0^h \in \mathcal{S}$ and the robot starts at $x_0^r \in \mathcal{S}$. Both agents are able to move in any direction at the same constant velocity V. The discrete-time dynamics of the human-robot team at each time step $k \geq 0$ are $x_{k+1}^h = x_k^h + u^h$ and $x_{k+1}^r = x_k^r + u^r$, respectively, where $\|u^h\| = \|u^r\| = V$, with $\|\cdot\|$ being the Euclidean norm. Both agents have the common objective of completing all tasks in the minimum amount of time. Each agent can observe its own location, the location of the other agent, and the location and type of all remaining tasks. However, agents cannot explicitly communicate with each other. This can be seen as a multi-agent version of the Traveling Salesman Problem [74].

Robot Collaboration Scheme

The scheme for integrating goal inference into task allocation is shown in Figure 6.2. It works as follows: At the initial time $k = 0$, the robot computes the optimal task allocation and capture sequence given the initial layout of tasks and agent positions (described in section "Task Allocation"). The robot observes the human's actions as both the human and robot move towards their respective goals. To infer whether the human is deviating from the robot's internal plan, the robot conducts a *maximum a posteriori* (MAP) estimate (described in section "Bayesian Goal Inference"). If a deviation is detected, the robot computes a new allocation of tasks based on the current state and the inferred human intent. The robot can achieve robustness to suboptimal human actions by adjusting its plan whenever the human's actions deviate from the robot's current internal allocation.

When computing the optimal allocation, it is assumed that the human will in fact execute the resulting joint plan. In the ideal scenario where both agents are perfectly rational, this would indeed be the case (without the need for explicit coordination). Of course, this rationality assumption is far from trivial in the case of human individuals; we nonetheless make use of it as an approximate model, admitting incorrect predictions and providing a mechanism to adapt in these cases.

It is worth noting that this model maximally exploits collaboration in those cases in which the prediction is correct: that is, if the human does follow the computed plan, then the team achieves the best possible performance at the task. As will be seen, adaptation based on this model leads to significant improvements in team performance.

Bayesian Goal Inference

Let x^g denote the location of any of the $N + M$ tasks. Once the human starts moving towards a new task, the robot infers the human's intent as the task with the highest posterior probability, $P(x^g|x^h_{0:k+1})$, given the history of observations thus far, $x^h_{0:k+1}$. The posterior probability is updated iteratively using Bayes' rule:

$$P(x^g|x^h_{0:k+1}) \;\propto\; P(x^h_{k+1}|x^g, x^h_{0:k})P(x^g|x^h_{0:k}),$$

where the prior $P(x^g|x^h_{0:k})$ corresponds to the robot's belief that the human is heading towards task x^g based on previous observed positions $x^h_{0:k}$. The likelihood function $P(x^h_{k+1}|x^g, x^h_{0:k})$ corresponds to the transition probability for the position of the human given a particular goal. Under the Markov assumption, the likelihood function can be simplified as:

$$P(x^h_{k+1}|x^g, x^h_{0:k}) = P(x^h_{k+1}|x^g, x^h_k).$$

We compute the term $P(x^h_{k+1}|x^g, x^h_k)$ assuming that the human is noisily optimizing some reward function. Following [9], we model this term as a Boltzmann ('soft-max') policy:

$$P(x^h_{k+1}|x^g, x^h_k) \;\propto\; \exp(\beta V_g(x^h_{k+1})), \tag{6.1}$$

where $V_g(x^h)$ is the value function for each task location x^g. We model $V_g(x^h)$ as a function of the distance between the human x^h and the task, $d_g(x^h) = \|x^g - x^h\|$; i.e. $V_g(x^h) = \gamma^{d_g(x^h)}U - c[(\gamma - \gamma^{d_g(x^h)})/(1-\gamma)]$. This corresponds to the value function of a γ-discounted optimal control problem with terminal reward U and uniform running cost c. The value function causes the Boltzmann policy to assign a higher probability to actions that take the human closer to the goal location x^g.

The parameter β is termed the *rationality coefficient*, capturing how diligent we expect the human to be in optimizing their reward. When $\beta = 0$, $P(x^h_{k+1}|x^g, x^h_k)$ gives a uniform distribution, with all actions equally likely independent of their reward (modeling an indifferent human agent). Conversely, as $\beta \to \infty$, the human agent is assumed to almost always pick the optimal action with respect to their current goal.

Task Allocation

We compute the optimal allocation and execution order of tasks that minimizes completion time using a mixed-integer linear program (MILP). Let $\mathcal{V}$ denote the set of human and robot agents and $\mathcal{G}$ represent the set of remaining tasks. Then, the task allocation problem can be formulated as:

$$\begin{aligned} \min_{\mathbf{y}_{alloc}, \mathbf{t}_{alloc}} \quad & \max_{v \in \mathcal{V}, g \in \mathcal{G}} t^v_g \\ \text{subject to} \quad & \mathbf{y}_{alloc} \in \mathbb{Y}_{feasible}, \quad \mathbf{t}_{alloc} \in \mathbb{T}_{feasible}, \end{aligned} \tag{6.2}$$

where t_g^v denotes the time when v^{th} agent completes task g; $\mathbf{y}_{alloc}$ encodes the allocation of tasks to each agent and $\mathbf{t}_{alloc}$ represents the time for agents to arrive and leave tasks; and $\mathbb{Y}_{feasible}$ and $\mathbb{T}_{feasible}$ denote the corresponding sets of feasible values that satisfy the requirements on completing one-agent and joint tasks. To generate an allocation that respects the intent of the human agent, the inferred goal is assigned as the first task for the human, which is enforced in $\mathbb{Y}_{feasible}$. Other constraints of $\mathbb{Y}_{feasible}$ include 1) each task is assigned to the human or robot, 2) an agent can only visit and complete a task once 3) an agent can leave a task position at most once and can only do so after visiting it (continuity condition) and 4) no subtour exists for any agent. The timing constraints of $\mathbb{T}_{feasible}$ requires that a joint task be completed the time when both the human and the robot visit the task while a common task be executed immediately when an agent visits it. The details of the MILP formulation of this particular traveling salesman problem is in Appendix A.4. In this study, the MILP is solved by the commercial solver Gurobi [68].

Evaluation

To rigorously test the robot's task allocation strategy and its effects on human-robot interaction, we ran a behavioral experiment in which we asked human participants to collaborate with three different robots, as shown in Figure 6.2a, one at a time and in a counter-balanced order: a robot that did not re-plan its strategy regardless of the human's choices (the *fixed* robot), a robot that reacted to these choices only once the human reached the goal in question (the *reactive* robot), and a robot that anticipated the human's next goal at each step and adapted its behavior early (the *predictive* robot). We conducted the experiment under two different conditions: one with an *oracle-predictive* robot that can make perfect inferences of the human's intent, and a second with a *Bayesian-predictive* robot, in which the Bayesian intent inference is at times erroneous (as it will be in real human-robot interactions).

6.2.3 Experimental Design

Task

We designed a web-based human-robot collaboration experiment where the human participant was in control of an avatar on a screen. In this experiment, the human and virtual robot had to complete a set of tasks as quickly as possible. While some of the tasks could be completed by either the human or the robot (*one-agent tasks*), others required both agents to complete the task simultaneously (*joint tasks*). The participant chose to complete tasks by clicking on them, causing the human avatar to move to that task while the robot moved towards a task of its choice. After the human completed the chosen task, they chose another task to complete, and so on until all tasks were completed.

Independent Variables

To evaluate the collaboration scheme described in Section 6.2.2, we manipulated two variables: **adaptivity** and **inference type**.

Table 6.1: Experimental conditions

Inference Type (between-subjects)	**Adaptivity** (within-subjects)	**Robot**
-	Low Adaptivity	*Fixed*
-	Med. Adaptivity	*Reactive*
Oracle	High Adaptivity	*Oracle-predictive*
-	Low Adaptivity	*Fixed*
-	Med. Adaptivity	*Reactive*
Bayesian	High Adaptivity	*Bayesian-predictive*

Adaptivity We manipulated the amount of information that the robot receives during the task, which directly affects how adaptive the robot is to human actions. The following robots vary from least adaptive to most adaptive:

Fixed. *This robot receives no information about the human's goal.* It runs an initial optimal task allocation for the human-robot team and subsequently executes the robot part of the plan irrespective of the human's actions. It only re-plans its task allocation if an inconsistency or deadlock is reached. For example, if the human finishes a task that the robot had internally allocated to itself, the fixed robot removes the task from its list of pending tasks and chooses the next pending task as the goal. Additionally, if both agents reach different joint tasks, the robot moves to the human's location and re-plans its strategy to account for this deviation.

Reactive. *This robot receives information about the human's goal only when the human reaches their goal.* At this point, the robot reruns task allocation for the team assuming that the human will behave optimally for the remainder of the trial.

Predictive. *This robot receives information about the human's goal before they reach the goal.* After receiving this information, the robot recomputes the optimal plan, again assuming optimal human behavior thereafter.

Inference Type We also manipulated the type of inference used by the predictive robot:

Oracle-predictive. *This robot receives perfect knowledge of the human's next goal as soon as the human chooses it.* Although there are real-world cases in which a human might inform a robot about its goal immediately after deciding, it is rare for a robot to know exactly the human's goal without having the human explicitly communicate it.

Bayesian-predictive. *This robot continuously receives information related to the human's goal by observing the motion of the human.* Based on this motion, the robot attempts to predict the human's goal using the Bayesian inference scheme described in Section 6.2.2. In most cases of human-robot collaboration, the robot will only have access to the human's motions.

Experimental Conditions

We manipulated the independent variables of adaptivity and inference type in a 3×2 mixed design in which inference type was varied between-subjects, while adaptivity was varied within-subjects. All participants collaborated with the fixed and reactive robots (corresponding to the *low adaptivity* and *medium adaptivity* conditions, respectively). For the *high adaptivity* conditions, half the participants were assigned to a *oracle* condition, in which they collaborated with the oracle-predictive robot; and half were assigned to a *Bayesian* condition, in which they collaborated with the Bayesian-predictive robot. This design is given in Table 6.1 for reference.

The motivation for this design was to prevent participants from having to keep track of too many robots (as would be the case in a 4×1 design), while still creating the experience of comparing the predictive robot with non-predictive baselines.

Procedure

Stimuli There were 15 unique task layouts which consisted of a collection of five or six tasks, as well as initial positions of the human and robot avatars. As shown in Figure 6.2a, one-agent tasks were depicted by an orange circle with a single dot inside it, and joint tasks were depicted by a cyan circle with two dots inside it. The human avatar was a gender-neutral cartoon of a person on a scooter, and the robot avatars were images of the same robot in different poses and colors (either red, blue, or yellow), which were counterbalanced across participants.

Trial Design First, we presented participants with four example trials to help them understand the task, which they completed with the help of a gray robot. Each experimental layout was presented to participants three times over the course of the experiment (once with each of the fixed, reactive, and predictive robots). The trials were grouped into five blocks of nine trials (45 trials total). Within each block, participants completed three trials with one of the robots, then three with a different robot, and then three with the final robot. The order of the robots and task layouts were randomized according to a Latin-square design, such that the order of the robots varied and such that no task layout was repeated within a block.

Layout Generation We randomly generated task layouts according to the following procedure. First, we assumed a model of human goal selection that sets the probability of choosing the next goal task as inversely proportional to the distance between the position of the human agent and the tasks[2]. Under this model, we calculated the expected task completion time of 104 randomly generated layouts with reactive and Bayesian-predictive robots. We then computed the rank ordering of these completion times relative to all possible completion times for each layout, and computed the ratio of the rank order of the Bayesian-adaptive robot to the rank order of the reactive robot. This ratio gave us a measure of how differently we would expect the two robots to behave: layouts

[2]Specifically, we used a Boltzmann policy to model human task selections. We fit the tuning parameter from participant's choices in an earlier pilot experiment using maximum-likelihood estimation, resulting in a best fitting parameter value of $\beta = 1.05$.

with ratio greater than 1.5 (9 layouts) or less than 0.6 (6 layouts) were selected, giving a total of 15 layouts for each robot. Subjects completed 5 blocks of 9 layouts, totaling 45 layouts.

Avatar Animation When moving to complete a task, the robot avatar always moved in a straight line towards the task. For the human avatar, however, the trajectory followed was not a straight path, but a Bézier curve that initially deviated to a side and then gradually corrected its course towards the correct task location. The reason for using this trajectory was to allow the Bayesian-predictive robot to make errors: if the human always moved in a straight line, then the Bayesian-predictive robot would almost always infer the correct goal right away (and thus would give nearly identical results as the oracle-predictive robot). Furthermore, this choice of curved trajectory is a more realistic depiction of how actual humans move, which is not usually in a straight line [21].

Attention Checks After reading the instructions, participants were given an attention check in the form of two questions asking them the color of the one-agent tasks and joint tasks. At the end of the experiment, we also asked them whether they had been paying attention to the difference in helpfulness between the three robots.

Controlling for Confounds We controlled for confounding factors by counterbalancing the colors of the robots; by using a different color robot for the example trials; by randomizing the trial order; by generating stimuli to give advantages to different robots; by animating the human avatar in a realistic way to differentiate between the Bayesian-predictive and oracle-predictive robots; and by including attention checks.

Dependent Measures

Objective measures During each trial, a timer on the right side of the screen kept track of the time it took to complete the tasks. This timer was paused whenever the participant was prompted to choose the next task and only counted up when either the human avatar or robot avatar were moving. The final time on the timer was used an object measure of how long it took participants to complete each trial. We also tracked how many one-agent tasks were completed by the robot, the human, or both simultaneously.

Subjective measures After every nine trials, we asked participants to choose the robot that they most preferred working with based on their experience so far. At the end of the experiment, we also asked participants to fill out a survey regarding their subjective experience working with each of the robot partners (Table 6.2). These survey questions were based on Hoffman's metrics for fluency in human-robot collaborations [78], and were answered on a 7-point Likert scale from 1 ("strongly disagree") to 7 ("strongly agree"). We also included two forced-choice questions, which had choices corresponding to the three robots.

Table 6.2: Subjective measures

Fluency
1. The human-robot team with the [color] robot worked fluently together.
2. The [color] robot contributed to the fluency of the team interaction.
Contribution
1. (Reversed) I had to carry the weight to make the human-robot team with the [color] robot better.
2. The [color] robot contributed equally to the team performance.
3. The performance of the [color] robot was an important contribution to the success of the team.
Trust
1. I trusted the [color] robot to do the right thing at the right time.
2. The [color] robot is trustworthy.
Capability
1. I am confident in the ability of the [color] robot to help me.
2. The [color] robot is intelligent.
Perceptiveness (originally 'Goals')
1. The [color] robot perceived accurately what my goals were.
2. (Reversed) The [color] robot did not understand what I was trying to accomplish.
Forced-Choice Questions
1. If you were to redo this experiment with only one of the robots, which would you most prefer to work with?
2. If you were to redo this experiment with only one of the robots, which would you least prefer to work with?

Hypotheses

We hypothesized that adaptivity and inference type would affect both objective measures and subjective measures.

H1 - Task Completion Time *Robot adaptivity will negatively affect task completion times, with the least adaptive robots resulting in the slowest completion times, and the most adaptive robots resulting in the fastest completion times. The Bayesian-predictive will robot result in slower completion times than the oracle-predictive robot.*

H2 - Robot Task Completion *Robot adaptivity will positively affect the number of tasks that the robot completes, with more adaptive robots completing more tasks than the less adaptive robots. The Bayesian-predictive robot will complete fewer tasks than the oracle-predictive robot.*

H3 - Robot Preference *Robot adaptivity will positively affect participants' preferences, with more adaptive robots being preferable to less adaptive robots. The Bayesian-predictive robot will be less preferable than the oracle-predictive robot. Additionally, preferences will become stronger as participants gain more experience with each robot.*

H4 - Perceptions of the Collaboration *Robot adaptivity will positively affect participants' per-*

ceptions of the collaboration as indicated by the subjective survey measures, with more adaptive robots being rated higher than less adaptive robots. Inference type will also affect perceptions, with the Bayesian-predictive robot scoring lower than the oracle-predictive robot.

Participants

We recruited a total of 234 participants from Amazon's Mechanical Turk using the psiTurk experimental framework [67]. We excluded 32 participants from analysis for failing the attention checks, as well as 6 due to an experimental error, leaving a total of $N = 196$ participants whose data we used. All participants were treated in accordance with local IRB standards and were paid \$1.20 for an average of 14.7 minutes of work, plus an average bonus of \$0.51. Bonuses could range from \$0.00 to \$1.35 depending on performance: for each trial, participants could get a \$0.03 bonus if they completed all tasks in the shortest possible amount of time; a \$0.02 bonus if they finished within the top 5% fastest times; or a \$0.01 bonus if they finished within the top 10% fastest times[3]. They received no bonus if they were slower than the top 10% fastest times.

6.2.4 Results

Manipulation Checks

The goal of changing the inference type was ultimately to influence the number of errors made by the robot. To ensure that the Bayesian-predictive robot did make more errors than the oracle-predictive robot, we counted the proportion of timesteps on each trial during which the Bayesian-predictive robot had an incorrect belief about the human's goal. We then constructed a mixed-effects model for these error rates with only the intercept as fixed effects, and both participants and stimuli as random effects. The intercept was 0.27 ± 0.021 SE ($t(14.043) = 12.83$, $p < 0.001$), indicating an average error rate of about 27%. We also looked at the stimuli individually by reconstructing the same model, except with stimuli as fixed effects. Using this model, we found that error rates ranged from 15.0% ± 1.18% SE to 41.0% ± 1.18% SE.

H1 - Task Completion Time

First, we looked at the objective measure of task completion time, which is also shown in Figure 6.3. To analyze task completion time, we first found the average completion time for each participant across stimuli, then constructed a linear mixed-effects model for these times using the inference type and adaptivity as fixed effects and the participants as random effects. We found main effects of both inference type ($F(2, 194) = 10.30$, $p < 0.01$) and adaptivity ($F(2, 388) = 709.27$, $p < 0.001$), as well as an interaction between the two ($F(2, 388) = 10.07$, $p < 0.001$).

We performed post-hoc comparisons using the multivariate t method for p-value adjustment. In support of **H1**, we found that the reactive robot led to completion times that were $0.52s \pm$

[3]When comparing completion times to the range of possible times, we always compared the participants' completion times with the range of all times achievable with the robot they were currently collaborating with.

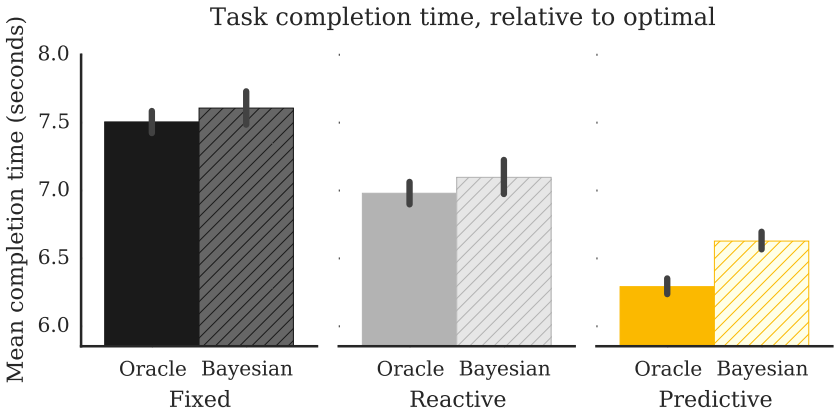


Figure 6.3: Comparison of completion times for different robots. The y-axis begins at the average optimal completion time, and thus represents a lower bound. Error bars are bootstrapped 95% confidence intervals of the mean using 1,000 samples.

0.029 SE faster than the fixed robot ($t(388) = 17.777$, $p < 0.001$). Also in support of **H1**, the oracle-predictive robot led to completion times that were $0.69s \pm 0.041$ SE faster than the reactive robot ($t(388) = -16.785$, $p < 0.001$), and the Bayesian-predictive robot led to completion times that were $0.47s \pm 0.041$ SE faster than the reactive robot ($t(388) = -11.339$, $p < 0.001$). There was no evidence for a difference between the fixed robots between participants ($t(329.53) = -1.528$, $p = 0.61$), nor between the reactive robots ($t(329.53) = -1.756$, $p = 0.46$). There was, however, a difference between the oracle-predictive and Bayesian-predictive robots, ($t(329.53) = -5.03$, $p < 0.001$), with the oracle-predictive robot resulting in completion times that were faster by $0.34s \pm 0.067$ SE. The combination of these results fully support **H1**: when adaptivity increases or the number of errors in inference decreases, task completion time decreases.

H2 - Robot Task Completion

Next, we looked at the additional object measure of how many one-agent tasks were completed by the robot.[4] For each participant, we counted the total number of one-agent tasks completed by the robot throughout the entire experiment. These numbers are shown in Figure 6.4. To analyze these results, we constructed a linear mixed-effects model for the number of tasks with inference type and adaptivity as fixed effects and the participants as random effects. There was no evidence for a main effect of inference type on the number of tasks completed by the robot ($F(2, 194) = 0.905$, $p = 0.34$), but there was an effect of the adaptivity on the number of tasks completed by the robot ($F(2, 388) = 171.15$, $p < 0.001$), as well as an interaction between adaptivity and inference type ($F(2, 388) = 9.237$, $p < 0.001$).

[4]Sometimes the robot and human could complete the same one-agent task simultaneously. We excluded these cases from the number of one-agent tasks completed by the robot, because for these tasks the robot was not actually providing any additional help. We found that the robot and human competed for the same task 1.5%, 8.6%, 14.6%, and 18.1% of the time in the oracle-predictive, Bayesian-predictive, reactive, and fixed conditions, respectively.

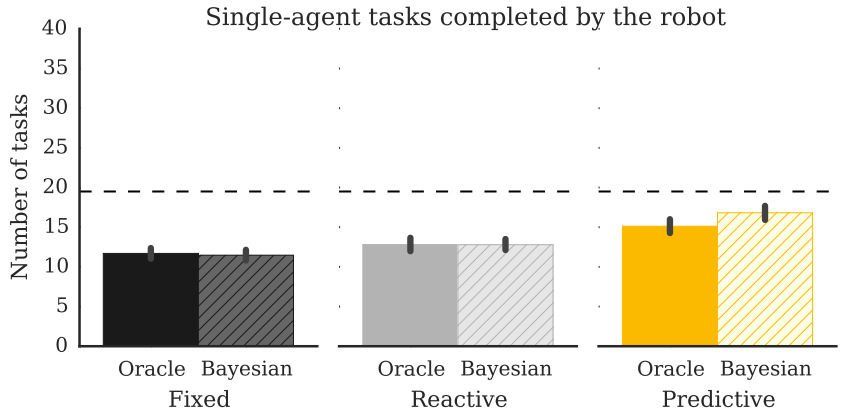


Figure 6.4: Number of one-agent tasks completed by the robot. The dotted lines represent the robot and the human completing an equal proportion of tasks. Error bars are bootstrapped 95% confidence intervals of the mean using 1,000 samples.

To investigate these effects further, we performed post-hoc comparisons with multivariate t adjustments. Consistent with **H2**, we found that the reactive robot completed 1.2 ± 0.24 SE more tasks than the fixed robot ($t(388) = 4.969$, $p < 0.001$); the oracle-predictive robot completed 2.3 ± 0.34 SE more tasks than the reactive robot ($t(388) = 6.722$, $p < 0.001$); and the Bayesian-predictive robot completed 4.0 ± 0.35 SE more tasks than the reactive robot ($t(388) = 11.566$, $p < 0.001$). Surprisingly, however, in comparing between subjects, we found the opposite of **H2**: the Bayesian-predictive robot completed 1.7 ± 0.58 SE more tasks than the oracle-predictive robot ($t(316.19) = 2.903$, $p < 0.05$).

These results confirm part of **H2**—that adaptivity increases the number of tasks completed by the robot—but disconfirm the other part of **H2**—that Bayesian inference decreases the number of tasks completed by the robot. However, we note that the effects are quite small between the different robots: the differences reported are in the order of 1–4 tasks for the entire experiment, which had a total of 39 one-agent tasks. Each trial only had two or three one-agent tasks, and thus these results may not generalize to scenarios that include many more tasks. Furthermore, although we found that the Bayesian inference type increased the number of tasks completed by the robot, we speculate that this may be due to the Bayesian-predictive robot inferring the incorrect human goal and therefore taking some other one-agent task. However, further investigation is required to determine the actual source of this difference.

H3 - Robot Preference

We next analyzed the subjective measure of participants' preferences for which robot they would rather work with.

Preferences over time Figure 6.5 shows the proportion of participants choosing each robot as a function of trial. We constructed a logistic mixed-effects model for binary preferences (where 1 meant the robot was chosen, and 0 meant it was not) with inference type, adaptivity, and trial as

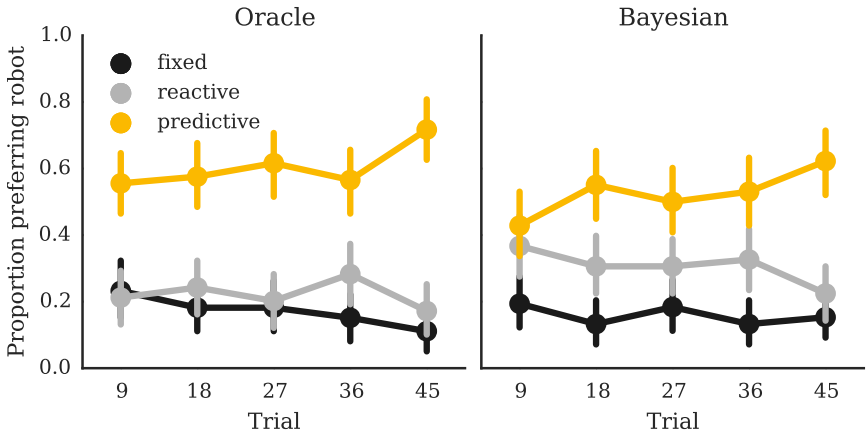


Figure 6.5: Preferred robot as a function of trial. Error bars are bootstrapped 95% confidence intervals of the mean using 1,000 samples.

fixed effects and participants as random effects. Using Wald's tests, we found a significant main effect of adaptivity ($\chi^2(2) = 16.7184$, $p < 0.001$) and trial ($\chi^2(1) = 5.1104$, $p < 0.05$), supporting **H3**. We also found an interaction between adaptivity and trial ($\chi^2(1) = 9.2493$, $p < 0.01$). While the coefficient for trial was negative ($\beta = -0.022 \pm 0.0095$ SE, $z = -2.261$, $p < 0.05$), the coefficient for the interaction of trial and the predictive robot was positive ($\beta = 0.036 \pm 0.0120$, $z = 3.017$, $p < 0.01$). This also supports **H3**: if preferences for the predictive robot increase over time, by definition the preferences for the other robots must decrease, thus resulting in an overall negative coefficient.

Although we did not find a significant main effect of inference type ($\chi^2(1) = 1.457$, $p = 0.23$), we did find an interaction between inference type and adaptivity ($\chi^2(2) = 7.7208$, $p < 0.05$). Post-hoc comparisons with the multivariate t adjustment indicated that there was a significant overall improvement in preference between the fixed and reactive robots ($z = -5.23$, $p < 0.001$) as well as an improvement in preference between the reactive and oracle-predictive robots ($z = 11.86$, $p < 0.001$) and between the reactive and Bayesian-predictive robots ($z = 6.95$, $p < 0.001$). While there was no significant difference between participants for the fixed robot ($z = 0.40$, $p = 0.998$), there was a significant difference for the reactive robot ($z = -2.93$, $p < 0.05$) and a marginal difference between the predictive robots ($z = 2.52$, $p = 0.098$), with the oracle-predictive robot being slightly more preferable to the Bayesian-predictive robot.

Final rankings For each participant, we assigned each robot a score based on their final rankings. The best robot received a score of 1; the worst robot received a score of 2; and the remaining robot received a score of 1.5. We constructed a logistic mixed-effects model for these scores, with inference type and adaptivity as fixed effects and participants as random effects; we then used Wald's tests to check for effects. In partial support of **H3**, there was a significant effect of adaptivity on the rankings ($\chi^2(2) = 52.352$, $p < 0.001$) but neither a significant effect of inference type ($\chi^2(1) = 1.149$, $p = 0.28$) nor an interaction between adaptivity and inference type ($\chi^2(2) = 3.582$, $p = 0.17$). Post-hoc comparisons between the robot types were adjusted with Tukey's HSD and indicated a preference for the reactive robot over the fixed robot ($z = 5.092$, $p < 0.001$) and a preference for the

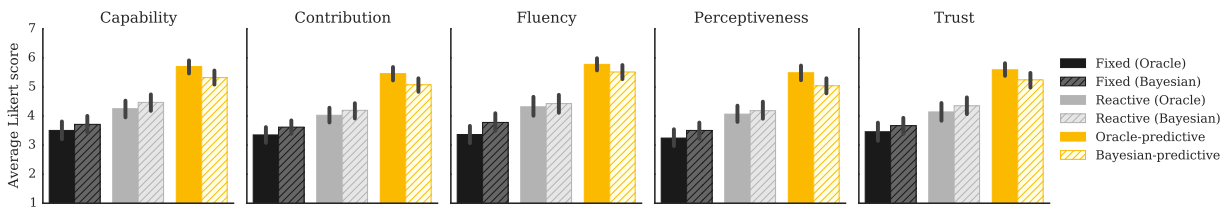


Figure 6.6: Findings for subjective measures. Error bars are bootstrapped 95% confidence intervals of the mean.

predictive robots over the reactive robot ($z = 5.634$, $p < 0.001$).

H4 - Perceptions of the Collaboration

Finally, we looked at the subjective measures of participants' perceptions of the collaboration. For each robot and measure (capability, contribution, fluency, goals, and trust), we averaged each participants' responses to the individual questions, resulting in a single score per participant, per measure, per robot. The results are shown in Figure 6.6. We constructed a linear mixed-effects model for the survey responses with the adaptivity, inference type, and measure type as fixed effects, and with participants as random effects. We found significant main effects for both the adaptivity ($F(2, 2730) = 543.18$, $p < 0.001$) and measure type ($F(4, 2730) = 5.28$, $p < 0.001$). While there was no significant main effect of inference type ($F(2, 195) = 0.08$, $p = 0.78$), there was an interaction between adaptivity and inference type ($F(2, 2730) = 17.17$, $p < 0.001$). We did not find interactions between inference type and measure type ($F(4, 2730) = 0.14$, $p = 0.97$), between adaptivity and measure type ($F(8, 2730) = 0.38$, $p = 0.93$), or between adaptivity, inference type, and measure type ($F(8, 2730) = 0.14$, $p = 0.997$).

We investigated these effects further through a post-hoc analysis with multivariate t corrections. In support of **H4**, we found that participants rated the reactive robot higher than the fixed robot by 0.72 ± 0.058 SE points on the Likert scale ($t(2730) = 12.415$, $p < 0.001$). They also rated the oracle-predictive robot higher than the reactive robot by 1.44 ± 0.082 SE points ($t(2730) = 17.568$, $p < 0.001$), the Bayesian-predictive robot higher than the reactive robot by 0.91 ± 0.083 SE points ($t(2730) = 11.064$, $p < 0.001$), and the oracle-predictive robot higher than the Bayesian-predictive robot by 0.36 ± 0.112 points ($t(470.31) = 3.267$, $p < 0.05$).

6.2.5 Discussion

This work proposes a task allocation approach that uses motion-level prediction to enable robot's adaptive task planning. An online behavior study was conducted, which sets out to answer three questions. First, does an increase in robot adaptivity lead to an objective improvement in real human-robot collaboration scenarios? Second, even if performance improves objectively, do people actually notice a difference? Third, if the robot does not have perfect knowledge of the human's goals and must infer them, does adaptivity still result in the same improvements?

Our results suggest that outfitting robots with the ability to adaptively re-plan their actions from a motion-based inference of human intent can lead to a systematic improvement in both the

objective and perceived performance in human-robot collaboration. We found that this type of adaptivity increases performance objectively by lowering the amount of time it takes the team to complete the tasks and by improving the balance of work between the two agents. Subjectively, we found that participants preferred motion-based predictive adaptation to purely reactive adaptation at the task level (which, as expected, was in turn preferred to no adaptation), and that they rated the more adaptive robot higher on a number of measures including capability, collaboration, fluency, perceptiveness, and trust. Unsurprisingly, introducing prediction errors through manipulation of the inference type generally decreased both objective performance and subjective ratings. What is surprising, however, is how small this decrease in performance was, particularly given that the Bayesian-predictive robot had an error rate ranging from a low of 15% to a high of 40% for some stimuli. This suggests that an attempt to predict and adapt to human goals—even under limited accuracy—can have a significant effect on the team's objective performance and the human's perception of the robot.

Another non-obvious observation is that the adaptation mechanism is successful in spite of the strong assumption that the human agent will behave optimally from the current state. In practice, the impact of this assumption is limited by the fact that the robot quickly adapts whenever the human does not follow this optimal plan; this is comparable to how model-predictive control can use a nominal system model and correct for model mismatch at each re-planning step. Additional effects may also explain the observed improvement. In particular, our model of human task planning explicitly accounts for the joint actions of the two agents (as opposed to only their own); because human inferences of other agent's intentions have been shown to be consistent with models of approximately rational planning [9], it is therefore likely that the robot's adapted actions closely match users' actual expectations. In addition, the fact that the robot adapts by switching to the new joint optimal plan means that the human is always "given a new chance" to behave optimally; in other words, the robot's adaptation is always placing the team in the situation with the best achievable outcome.

6.3 Planning for Predictability[5]

6.3.1 Introduction

The aim in this work is to facilitate human-robot collaboration by eliminating the misalignments between what the automated agent is planning to do and what the human believes it is going to do: we want humans to be able to infer a robot's plan during a collaborative task by observing the robot's actions. Previous works usually focus on the motion-level planning for intent conveyance about a single goal. In this work, we have considered the predictability of a plan on task level, which consists of multiple subgoals. Especially, we focus on conveying robots' high-level plans (action sequences) to humans in order to facilitate collaboration in mixed human-robot teams.

This work introduces a framework that explicitly predicts the inferences that human observers are likely to make for the robot's plan based on its initial actions, and incorporates them into the planning process in order to generate plans that are more easily and unambiguously interpretable

[5]This section is based on the work published in [52].

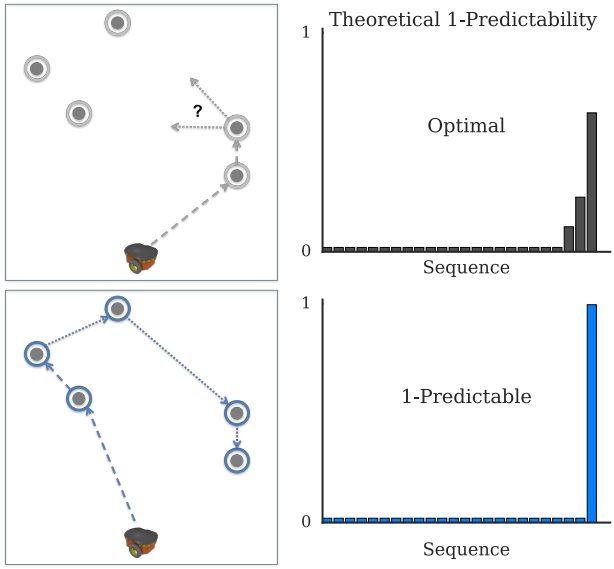


Figure 6.7: Left: Schematic of the trajectory followed by the optimal (top) and 1-predictable (bottom) planner. Right: Theoretical 1-predictability of candidate sequences after the robot has visited a single target.

by human collaborators. Such framework relies on the connection between models of human planning and that of human inference that was proposed in cognitive science [9]. It models the human as a noisy observer who will infer an agent's plan with exponentially decaying probability as its cost increases, which we expect that a similar mechanism will also apply to many peer-to-peer collaboration scenarios.

This work focuses on communicating the *sequences of future actions* from the *initial actions* and a *known goal*. Such situations usually appear in *task planning*, where the overall goal of the task is clear (completing all subgoals), but the sequence of actions the robot will take to achieve the goal is not. The key difference between this work and previous works is the richness of information that needs to be conveyed. Prior works have developed algorithms for generating trajectories that communicate the robot's *overall goal* from the ongoing *motion* [156, 46, 155, 62] while this work conveys the sequences of remaining actions.

This work makes an initial effort in defining a framework that explicitly accounts for human's inference of robot's plan, and generating robot plans that are unambiguously intelligible by human partners. Specifically, we define a property of a robot plan that we refer to as *t-predictability*: a plan is t-predictable if a human can infer the robot's remaining actions in a task from having observed only the first t actions and from knowing the robot's overall goal in the task. We then propose a planning algorithm that generates t-predictable plans, that is, the plans that a human can easily infer the rest of actions from the initial ones. We test the effects of t-predictable plans on human's inference via an extensive online user study and investigate the implications on human-robot collaboration via an in-person study.

6.3.2 Conveying Intent via Planning of Action Sequences

In this section, we consider the conveyance of robot intent to a human observer via planning of action sequences. Figure 6.7 gives an example. Different from traditional works that unambiguously express an agent's *goal* via the design of motion trajectory, our work focuses on that, when the *goal* is given, the whole action sequence is clearly conveyed when only the partial sequence is shown. To achieve this, we propose the concept of t-predictability and formulate the planning of t-predictable sequences as an optimization problem over the space of all possible plans.

Definition of t-predictability

We consider a task planning problem from a starting state $S \in \mathcal{S}$ with an overall goal $G \in \mathcal{G}$ that can be achieved through a series of actions, called a *plan*, within a finite horizon T. To quantitatively describe the property of a sequence in terms of the easiness of inferring an agent's plan, we define the concept of t-predictability .

Definition 2 *(t-predictability) Let $\mathcal{U}$ denote the space of all feasible plans of length (up to) T that achieve the goal. The t-predictability $\mathcal{P}_t$ of a feasible plan $\mathbf{u} = [u_1, u_2, ..., u_T] \in \mathcal{U}$ that achieves an overall goal G is the probability of an observer correctly inferring $[u_{t+1}, ..., u_T]$ after observing $[u_1, ..., u_t]$, and knowing the overall goal G. Specifically, this is given by*

$$\mathcal{P}_t(\mathbf{u}) = P(u_{t+1}, ..., u_T | S, G, u_1, ..., u_t).$$

Definition 3 *(t-predictable planner) A t-predictable planner generates the plan that maximizes t-predictability out of all those that achieve the overall goal G. That is, a t-predictable planner generates the action series $\mathbf{u}^*$ such that $\mathbf{u}^* = \arg\max_{\mathbf{u}\in\mathcal{U}} \mathcal{P}_t(\mathbf{u})$.*

This is equivalent to:

$$\mathbf{u}^* = \arg\max_{\mathbf{u}\in\mathcal{U}} \frac{P(u_1, ..., u_T | S, G)}{\sum_{[\tilde{u}_{t+1}, ..., \tilde{u}_T]} P(u_1, ..., u_t, \tilde{u}_{t+1}, ..., \tilde{u}_T | S, G)}. \tag{6.3}$$

Remark 4 *t-predictability is a measure of the degree of confidence with which an observer will accurately infer a plan, which means we can quantitatively computes the t-predictability score of any sequence. The t-predictable planner returns the action sequence with highest t-predictability score among all possible sequences. It is possible that highly t-predictable sequence may not exist in some cases, which means there exist at least two sequences with close t-predictability score. However, in many other cases, highly t-predictable sequence exists and optimizing for t-predictability is highly useful, as evidenced by our experiment results.*

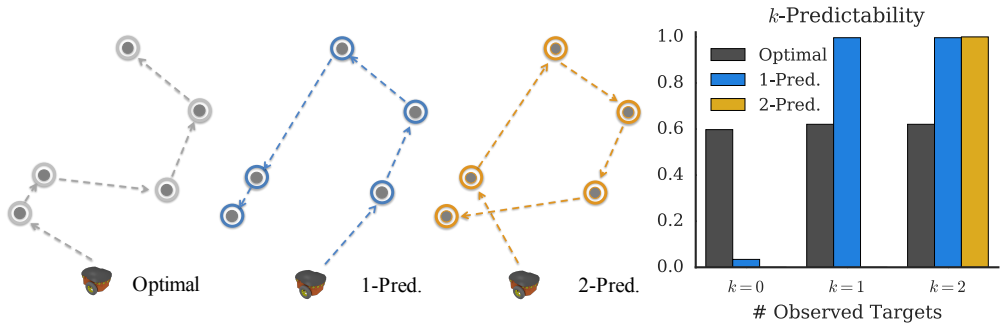


Figure 6.8: Theoretical t-predictability. Left: sequences generated by the three planners for a typical task layout. Right: theoretical k-predictability of the same three sequences under different numbers k of observed targets. In all cases, the highest value corresponds to $t = k$.

Illustrative Example

Fig. 6.8 shows the outcome of optimizing for t-predictability in a Traveling Salesman context, with $t = 0$, 1, and 2 targets, and considers the theoretical k-predictability for each plan, with k the number of *actually* observed targets (which may be different from the t assumed by the planner). The 0-predictable plan (gray, left) is the best when the observer sees no actions, since it is the optimal plan. However, it is no longer the best plan when the observer gets to see the first action: whereas there are multiple low-cost remaining sequences after the first action in the 0-predictable plan, there is only one low-cost remaining sequence after the first action in the 1-predictable plan (blue, center). The first action in the 2-predictable (orange, right) seems irrational, but this plan is optimized for when the observer gets to see the first two actions: indeed, after the first two actions, the remaining plan is maximally clear.

Relation to Predictability

t-predictability generalizes predictability [43]. For $t = 0$, the t-predictability of a plan simply becomes its predictability, that is, the ease with which the entire sequence of actions can be inferred with knowledge of the overall goal G, i.e. $\mathcal{P}_0 = P(a_1, .., a_T|S, G)$.

Relation to Legibility

Legibility [46] as applied to task planning would maximize the probability of the goal given the beginning of the plan, i.e. $P(G|S, a_1, .., a_t)$. In contrast, with t-predictability the robot is given a high-level goal describing some state that the world needs to be brought into (for example, clearing all objects from a table), and the observer is *already* aware of this goal. Instead of communicating the goal, the robot conveys the remainder of the plan using the first few elements, maximizing $P(a_{t+1}, .., a_T|S, G, a_1, .., a_t)$.

One important implication is that for t-predictability, unlike for legibility, *there is no a-priori set entity to be conveyed.* The algorithm searches for *both* a beginning and a remainder of a plan such

that, by observing the former, the observer can correctly guess the latter. Furthermore, legibility and t-predictability entail a different kind of information encoding: in legibility, the robot uses a partial trajectory or action sequence to indicate a single goal state, whereas in t-predictability the robot uses a partial action sequence to indicate *another* action sequence. Therefore, one entails a mapping from a large space to a small set of possibilities (the finite candidate goal states), whereas the other entails a mapping between spaces of equivalent size.

The distinction between task-level legibility and t-predictability is crucially important, particularly in collaborative settings. If you are cooking dinner with your household robot, it is important for the robot to act legibly so you can infer *what* goal it has when it grabs a knife (e.g., to slice vegetables). But, it is equally important for the robot to act in a t-predictable manner so that you can predict *how* it will accomplish that goal (e.g., the order in which it will cut the vegetables).

Relation to Explicability.

Explicability [183] has been recently introduced to measure whether the observer could assign labels to a plan. In this context, explicability would measure the existence of any remainder of a plan that achieves the goal, as opposed to optimizing the probability that the observer will infer the robot's plan.

Boltzmann Noisy Rationality

Similar to Section 6.2, this work also utilizes the Boltzmann probabilistic models of human noisy optimality. It assumes that the human observer expects the other agent to be noisily optimal in taking actions to achieving a goal. We adopt an analogous model for modeling the inference of *action sequences*: the human is now modeled as expecting the robot to be noisily optimal, taking approximately the optimal sequence of actions to achieve G. This is actually consistent with recent cognitive research on human understanding of complex plans [127], although our approach takes a higher-level perspective and considers each step in the overall plan as the basic action.

We define optimality via some cost function $c : \mathcal{U} \times \mathcal{S} \times \mathcal{G} \rightarrow \mathbb{R}^+$, mapping each action sequence for a particular goal and from a starting state to a scalar cost. In this work, we use path length (travel distance) for c. Applying the Bolzmann policy [9] based on c, we get:

$$P(\mathbf{u}|S,G) = \frac{e^{-\beta c(\mathbf{u},S,G)}}{\sum_{\tilde{\mathbf{u}}\in\mathcal{U}} e^{-\beta c(\tilde{\mathbf{u}},S,G)}}. \tag{6.4}$$

Similar to Eq. (6.1), $\beta > 0$ is the rationality coefficient. As $\beta \rightarrow \infty$ the probability distribution converges to one for the optimal sequence and zero elsewhere, corresponding to the case of a rational agent. As $\beta \rightarrow 0$, the probability distribution becomes uniform over all possible sequences $\mathbf{u}$ and the agent is indifferent; in such case, no intent information can be obtained by merely observing the agent's actions.

Remark 5 *The method proposed in this section is a general framework for generating predictable paths. We choose the TSP scenario since it is a classical problem that people are familiar with. In*

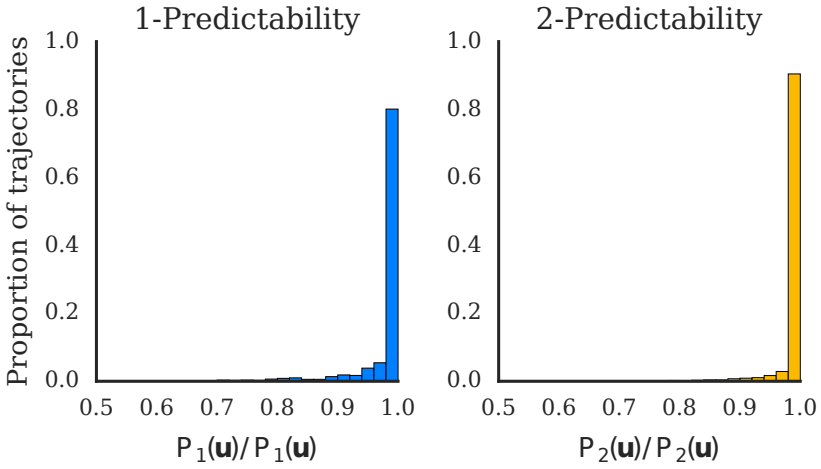


Figure 6.9: Approximate t-predictability. Each subplot shows a histogram of ratios between exact t-predictability ($\mathcal{P}_t$) and approximate t-predictability ($\widetilde{\mathcal{P}}_t$) of all possible sequences across 270 unique layouts. The subplots show histograms of ratios for 1- and 2-predictability, respectively. In the majority of cases, the exact and approximate t-predictabilities are nearly identical.

fact, humans are known to perform remarkably well at the TSP for as many as 20 targets [119]. Therefore we expect that humans can nicely infer the robot's plan by conducting mental simulation. Using TSP also allows us to isolate and measure the effects of t-predictability without confounding factors arising from structural complexity.

t-Predictability Optimization

We make the assumption that cost is linearly separable, i.e. $c(\mathbf{u}, S, G) = \sum c(u_t, S^t_{\mathbf{u}}, G)$. The solution to Eq. (6.3) can be computed by enumeration in problems with small T. For large T, an approximation algorithm is presented in the next part. Incorporating Eq. (6.4) into Eq. (6.3), it follows that

$$\mathbf{u}^* = \arg\max_{\mathbf{u}\in\mathcal{U}} \frac{\exp\big(-\beta c([u_{t+1}, ..., u_T], S^t_{\mathbf{u}}, G)\big)}{\sum_{[\tilde{u}_{t+1},...,\tilde{u}_T]\in\mathcal{U}^t_{\mathbf{u}}} \exp\big(-\beta c([\tilde{u}_{t+1}, ..., \tilde{u}_T], S^t_{\mathbf{u}}, G)\big)} \tag{6.5}$$

with $S^t_{\mathbf{u}}$ denoting the state reached by executing the first t steps of plan $\mathbf{u}$, and $\mathcal{U}^t_{\mathbf{u}}$ denoting the set of all feasible plans that achieve G from state $S^t_{\mathbf{u}}$ in $T - t$ steps or less.

Approximate Algorithm for Large-Scale Optimization.

The challenge with the optimization in (6.5) is the denominator: it requires summing over all possible plan remainders. Motivated by the fact that plans with higher costs contribute exponentially less to the sum, we propose to approximate the denominator by only summing over the lowest-cost l plan remainders.

Many tasks have the structure of Traveling Salesman Problems (TSP), where there are a number of subgoals whose order is not constrained but influences total cost. Van der Poort et al. [165] showed how to efficiently compute the l best solutions to the standard (cyclic tour) TSP using a branch-and-bound algorithm. The key mechanism is successively dividing the set of feasible plans into smaller subsets for which a lower bound on the cost can be computed by some heuristic. When a subset of solutions has a lower bound higher than the smallest l costs evaluated so far, it is discarded entirely, while the remaining subsets continue to be broken up. The process continues until only l feasible plans remain. This method is guaranteed to produce the l solutions with the least cost and can significantly reduce time complexity over exhaustive enumeration. In particular, it has been shown that for the standard TSP, computation is in $O(l(T-t)^3 2^{T-t})$ [165], while exhaustive enumeration requires computation in $O((T-t)!)$. While heuristics are domain-specific, we expect that this method can be widely applicable to robot task planning problems. Further, we expect $T-t$ to be a small number in realistic applications, limited by people's ability to reason about long sequences of actions.

To empirically evaluate the consequences of this approximation of t-predictability, we computed the exact and approximate (using $l = 2$) t-predictability for all possible plans in 270 randomly generated unique scenes (Fig. 6.9). If we choose the maximally t-predictable sequences for each scene using both the exact and approximate calculations of t-predictability, we find that these sequences agree in 242 (out of 270) scenes for 1-predictability and in 263 for 2-predictability[6]. For the sequences that disagree, the exact t-predictability of the sequence chosen using the approximate method is 89.5% of the optimal t-predictability in the worst case, and 99% of the optimal t-predictability on average. This shows that the proposed approximation is highly effective at producing t-predictable plans.

6.3.3 Evaluation via Online Experiment

We set up an experiment to test that our t-predictable planner is in fact t-predictable. We designed a web-based virtual human-robot collaboration experiment in a TSP setting (Figure 6.10), where the human had to predict the behavior of three robot avatars using different planners. Participants watched the robots move to a number of targets (either zero, one, or two) and had to predict the sequence of remaining targets the robot would complete. The experiment was conducted via the Amazon Mechanical Turk using the psiTurk experimental framework [67]. We describe the setup of the experiment and analyze the results in the following sections.

Experiment Setup

Independent Variables We manipulated two variables: the t-predictable planner (for $t \in \{0,\ 1,\ 2\}$) and the number of observed targets k (for $k \in \{0,\ 1,\ 2\}$).

Planner. We used three different planners which differed in their optimization criterion: the number t of targets assumed known to the observer. Each participant interacted with three robot avatars, each using one of the following three planners:

[6]For 0-predictability, the denominator is the same in all plans, so all 270 scenes agree trivially.

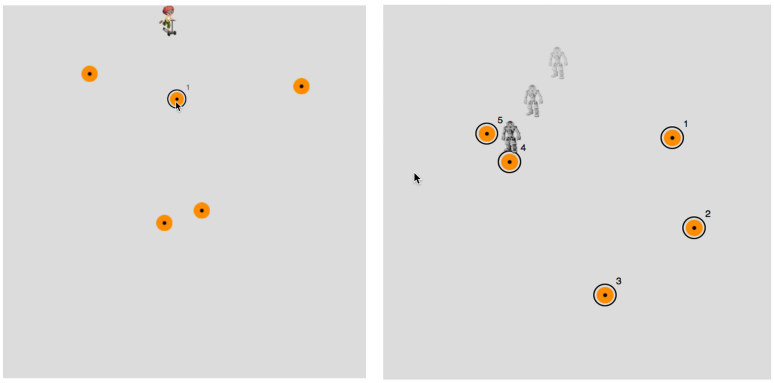


Figure 6.10: Examples of training and experiment phase. Left: participants can click on a target and the human avatar will move to it. Right: participants first predict action sequence (numbers indicating the order), then the robot avatar moves to these targets showing its actual sequence.

Optimal (0-predictable): This robot chooses the shortest path from the initial location visiting all target locations once; that is, the "traditional" solution to the open TSP. This robot solves (6.5) for $t = 0$.

1-predictable: This robot solves (6.5) for $t = 1$; the sequence might make an inefficient choice for the first target in order to make the sequence of remaining targets clear.

2-predictable: This robot solves (6.5) for $t = 2$; the sequence might make an inefficient choice for the first *two* targets in order to make the sequence of remaining targets clear.

Number of observed targets. Each subject was shown the first $k \in \{0, 1, 2\}$ targets of the robot's chosen sequence in each trial and was asked to predict the remainder of the sequence. This variable was manipulated between participants; thus, a given participant always saw the same number k of initial targets on all trials.

Procedure The experiment was divided into two phases: a training phase to familiarize participants with TSPs and how to solve them, and an experimental phase. We additionally asked participants to fill out a survey at the end of the experiment.

In the training phase, subjects controlled a human avatar. They were instructed to click on targets in the order that they believed would result in the quickest path for the human avatar to visit all of them. The human avatar moved in a straight line after each click and "captured" the selected target, which was then removed from the display.

For the second phase of the experiment, participants saw a robot avatar move to either $k = 0$, $k = 1$, or $k = 2$ targets. After moving to these targets, the robot paused so that participants could predict the remaining sequence of targets by clicking on the targets in the order in which they believed the robot would complete them. Afterwards, participants were presented with an animation showing the robot moving to the rest of the targets in the sequence determined by the corresponding planner.

Stimuli. Each target layout displayed a square domain with five or six targets. There were a total of 60 trials, consisting of four repetitions of 15 unique target layouts in random order: one for the training phase, in addition to the three experimental conditions. The trials were grouped so that each participant observed the same robot for three trials in a row before switching to a different robot. In the training trials, the avatar was a gender-neutral cartoon of a person on a scooter, and the robot avatars were images of the same robot in different poses and colors (either red, blue, or yellow).

Layout Generation. The layouts for the 15 trials were based from an initial database of 270 randomly generated layouts. This number was reduced down to 176 in which the chosen sequence was different between all three planners so that the stimuli were distinguishable. We also discarded some scenarios in which the robot's trajectory approached a target without capturing it, to avoid confounds. Out of these valid layouts, we chose the ones with the highest theoretical gain in 1-predictability to 2-predictability, to avoid scenarios where the information gain was marginal.

Attention Checks. After reading the instructions, participants were given an attention check in the form of two questions asking them the color of the targets and the color of the robot that they *would not* be evaluating.

Controlling for Confounds. We controlled for confounds by counterbalancing the colors of the robots for each planner; by using a human avatar in the practice trials; by randomizing the trial order; and by including attention checks.

Dependent Measures

Objective measures. We recorded the proportion of correct predictions of the robot's sequence of targets out of all 15 trials for each planner, resulting in a measure of *error rate*. We additionally computed the *Levenshtein distance* between predicted and actual sequences of targets. This is a more precise measure of how similar participants' predictions were to the actual sequences produced by the planner.

Subjective measures. After every ninth trial of the experiment, we asked participants to indicate which robot they preferred working with. At the end of the experiment, each participant was also asked to complete a questionnaire (adapted from [42]) to evaluate their perceived performance of three robots. The questionnaire, as shown in Table 6.3, consists of questions evaluating human's perception of the robots in terms of t-predictability, consistency, capability and helpfulness. A 7-level Likert scale was used to represent degrees from "Strongly agree" to "Strongly disagree". There was an open question (optional) at the end for participants to leave comments about their feelings of the robot.

Since each participant only experience a fixed k, the number of observed targets, in all trials, different questions were designed for k and presented to corresponding people. These is shown in questions numbered 1.x and 2.x in"t-predictability" and "Consistency" categories in Table 6.3. In addition, "Consistency" questions were only asked to participants who saw either one or two targets ($k = 1$ or 2). The "(color)" was replaced by the corresponding robot's color in the actual questionnaire given to participants.

Hypotheses

Table 6.3: Subjective Measures

***t*-predictability**
1.0 The (color) robot's sequence of tasks is easy to predict.
1.1 The (color) robot's sequence of tasks is easy to predict after seeing the first task.
1.2 The (color) robot's sequence of tasks is easy to predict after seeing the first two tasks.
2.0 I find it hard to anticipate the order in which the (color) robot will complete the tasks.
2.1 I find it hard to guess the rest of the (color) robot's sequence after seeing it do the first task.
2.2 I find it hard to guess the rest of the (color) robot's sequence after seeing it do the first two tasks.
Consistency
1.1 The (color) robot behaves in a coherent way for the first task.
1.2 The robot behaves in a coherent way for the first two tasks.
2. The start of the (color) robot's sequence (before my prediction) does not make any sense.
Capability
1. The (color) robot seems to know what it's doing.
2. The (color) robot chose its actions poorly.
Helpfulness
1. Overall, I feel that the robot's choice of sequence made my prediction job easier.
2. The robot's choices were confusing and made my prediction job less straightforward.

H1 - Comparison with Optimal. *When showing 1 target, the* 1*-predictable robot will result in lower error than the optimal baseline. When showing 2 targets, the* 2*-predictable robot will result in lower error than the optimal baseline.*

H2 - Generalization. *The error rate will be lowest when t = k: the number of targets shown, k, equals the number of targets assumed by the t-predictable planner, t.*

H3 - Preference. *The perceived performance of the robots will be highest when t = k.*

Participants We recruited a total of 242 participants from Amazon's Mechanical Turk using the psiTurk experimental framework [67]. We excluded 42 participants from analysis for failing the attention checks, leaving a net total of $N = 200$ participants. All participants were treated in accordance with local IRB standards and were paid \$1.80 USD for an average of 22 minutes of work, plus an average performance-based bonus of \$0.47.

Results

Model validity We first looked at the validity of our model of t-predictability with respect to people's performance in the experiment. We computed the theoretical k-predictability (probability of correctly predicting the robot's sequence from the k targets the user saw) for each task layout

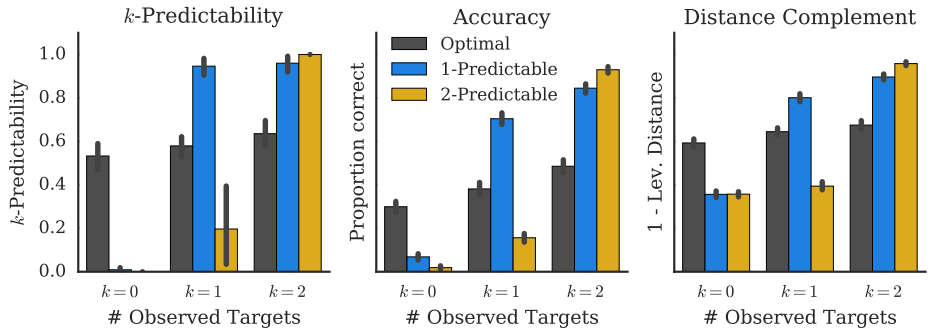


Figure 6.11: Predictability, error rate and edit distance. Left: theoretical k-predictability of sequences generated by different t-predictable planners under different numbers k of observed targets, averaged over all task layouts used in the online experiment. In all cases, the highest value corresponds to $t = k$. Center: empirical proportion of correct predictions with different t-predictable planners for different numbers k of observed targets. Right: complement of the average empirical Levenshtein distance between predicted and correct sequences. The lowest experimental error rates under both metrics occur when $t = k$.

under each planner and number of targets the users observed. We also computed people's actual prediction accuracy on each of these layouts under each condition, averaged across participants.

We computed the Pearson correlation between k-predictability and participant accuracy, finding a correlation of $r = 0.87$, 95% CI $[0.81, 0.91]$; the confidence interval around the median was computed using 10,000 bootstrap samples (with replacement). This high correlation suggests that our model of how people predict action sequences of other agents is a good predictor of their actual behavior.

Accuracy To determine how similar people's predictions of the robots' sequences were to the actual sequences, we used two objective measures of accuracy: first, overall error rate (whether they predicted the correct sequence or not), as well as the Levenshtein distance between the predicted and correct sequences (Figure 6.11).

As the two measures have qualitatively similar patterns of results, and the Levenshtein distance is a more fine-grained measure of accuracy, we performed quantitative analysis only on the Levenshtein distance. We constructed a linear mixed-effects model with the number of observed targets k (k from 0 to 2) and the planner for t-predictability (t from 0 to 2) as fixed effects, and trial layout as random effects.

This model revealed significant main effects of the number of observed targets ($F(2, 10299) = 1894.75$, $p < 0.001$) and planner ($F(2, 42) = 6.59$, $p < 0.01$) as well as an interaction between the two ($F(4, 10299) = 554.00$, $p < 0.001$). We ran post-hoc comparisons using the multivariate t adjustment. Comparing the planners across the same number of targets, we found that in the 0-targets condition the optimal (or 0-predictable) robot was better than the other two robots; in the 1-target

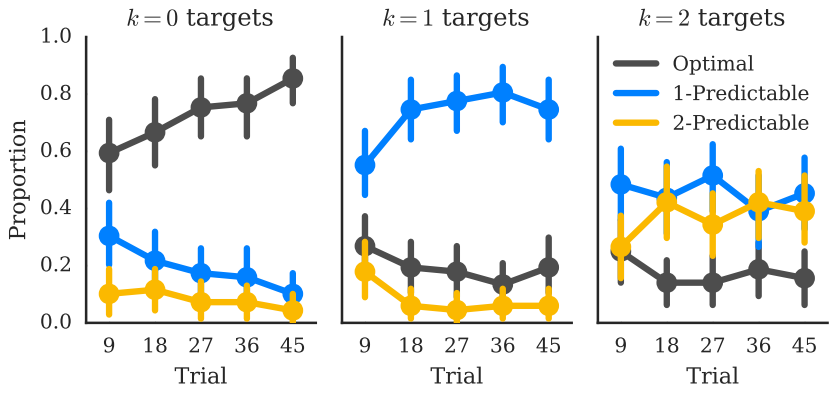


Figure 6.12: Preferences over time. Participants prefer the 0-predictable (optimal) robot for $k = 0$ and the 1-predictable robot for $k = 1$, as well as $k = 2$ (despite performing better with the 2-predictable robot, subjects often report being confused or frustrated by its first 2 actions.)

condition, the 1-predictable robot was better than the other two; in the 2-target prediction, the 2-predictable robot was better than the optimal and 1-predictable robots. All differences were significant with $p < 0.001$ except the difference between the 2-predictable robot and the 1-predictable robot in the 2-target condition ($t(50) = 1.85,\ p = 0.56$). Comparing the performance of a planner across number of targets, we found significant differences in all contrasts, with one exception: the accuracy when using the optimal planner was not significantly different when seeing 1 target vs 2 targets ($t(10299) = 2.65,\ p = 0.11$). Overall, these results support our hypotheses **H1** and **H2**, that accuracy is highest when t used in the planner equals k, the number of observed targets.

Preferences over time Figure 6.12 shows the proportion of participants choosing each robot planner at each trial. We constructed a logistic mixed-effects model for binary preferences (where 1 meant the robot was chosen) with planner, number of observed targets, and trial as fixed effects and participants as random effects. The planner and number of observed targets were categorical variables, while trial was a numeric variable.

Using Wald's tests, we found a significant main effect of the planner ($\chi^2(2) = 13.66, p < 0.01$) and trial ($\chi^2(1) = 9.30, p < 0.01$). We detected only a marginal effect of number of targets ($\chi^2(2) = 4.67, p = 0.10$). However, there was a significant interaction between planner and number of targets ($\chi^2(4) = 20.26, p < 0.001$). We also found interactions between planner and trial ($\chi^2(2) = 24.68, p < 0.001$) and between number of targets and trial ($\chi^2(2) = 16.07, p < 0.001$), as well as a three-way interaction ($\chi^2(4) = 39.43, p < 0.001$). Post-hoc comparisons with the multivariate t adjustment for p-values indicated that for the 0-targets condition, the optimal robot was preferred over the 1-predictable robot ($z = -13.22,\ p < 0.001$) and the 2-predictable robot ($z = -14.56,\ p < 0.001$). For the 1-target condition, the 1-predictable robot was preferred over the optimal robot ($z = 12.97,\ p < 0.001$) and the 2-predictable robot ($z = 14.00,\ p < 0.001$). In the two-task condition, we did not detect a difference between the two 1-predictable and 2-predictable

robots ($z = 2.26,\ p = 0.29$), though both were preferred over the optimal robot ($z = 7.44,\ p < 0.001$ for the 1-predictable robot and $z = 5.40,\ p < 0.001$ for the 2-predictable robot).

Overall, these results are in line with our hypothesis **H3** that the perceived performance is highest when t used in the planner equals k, the number of observed targets. This is the case for $k = 0$ and $k = 1$, but not $k = 2$: even though users tended to perform better with the 2-predictable robot, its suboptimal actions in the beginning seemed to confuse and frustrate users (see Qualtitative feedback results for details).

Final rankings The final rankings of "best robot" and "worst robot" are shown in Figure 6.13. For each participant, we assigned each robot a score based on their final rankings. The best robot received a score of 1; the worst robot received a score of 2; and the remaining robot received a score of 1.5. We constructed a logistic mixed-effects model for these scores, with planner and number of observed targets as fixed effects, and participants as random effects; we then used Walds tests to check for effects.

We found significant main effects of planner ($\chi^2(2) = 41.38, p < 0.001$) and number of targets ($\chi^2(2) = 12.97, p < 0.01$), as well as an interaction between them ($\chi^2(4) = 88.52, p < 0.001$). We again performed post-hoc comparisons using the multivariate t adjustment. These comparisons indicated that in the 1-target condition, people preferred the optimal robot over the 1-predictable robot ($z = 3.46,\ p < 0.01$) and the 2-predictable robot ($z = 5.60,\ p < 0.001$). In the 1-target condition, there was a preference for the 1-predictable robot over the optimal robot, however this difference was not significant ($z = -2.18,\ p = 0.27$). The 1-predictable robot was preferred to the 2-predictable robot ($z = -6.54,\ p < 0.001$). In the 2-target condition, both the 1-predictable and 2-predictable robots were preferred over the optimal robot ($z = -4.85,\ p < 0.001$ for the 1-predictable robot, and $z = -3.85,\ p < 0.01$ for the 2-predictable robot), though we did not detect a difference between the the 1-predictable and 2-predictable robots themselves ($z = -1.33,\ p = 0.84$). Overall, these rankings are in line with the preferences over time.

Qualitative feedback At the end of the experiment, we asked participants to briefly comment on each robot. For $k = 0$, responses typically favored the optimal robot, often described as "efficient" and "logical", although they also showed some reservations: *"close to what I would do but just a little bit of weird choices tossed in"*. Conversely, for $k > 0$, the optimal robot was likened to "a dysfunctional computer", and described as "ineffective" or "very robotic": *"I feel like maybe I'm a dumb human and the [optimal] robot might be the most efficient, because I have no idea. It frustrated me."*

The 2-predictable robot had mixed reviews for $k = 2$: for some it was "easy to predict", others found it "misleading" or noted its "weird starting points". For $k < 2$, it was reported as "useless", "all over the place", and *"terribly unintuitive with an abysmal sense of planning"*; one participant wrote it *"almost seemed like it was trying to trip me up on purpose"* and another one declared *"I want to beat this robot against a wall."*

The 1-predictable robot seemed to receive the best evaluations overall: though for $k = 0$ many users found it "random", "frustrating" and "confusing", for $k > 0$ it almost invariably had positive reviews ("sensible", "reasonable", "dependable", "smart", "on top of it"), being likened to "a

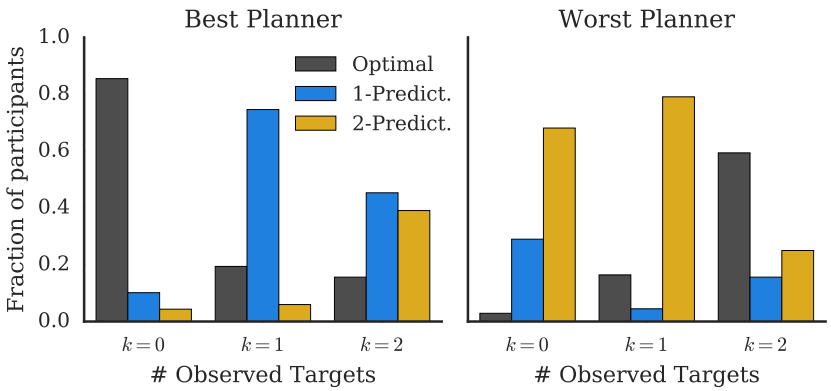


Figure 6.13: Final rankings. Participants ranked the planners differently depending on how many targets were observed. For $k = 0$, people preferred the optimal planner; for $k = 1$ and $k = 2$, they preferred the 1-predictable planner.

logical and rational human" and even eliciting positive emotions: *"You're my boy, Blue!"*, *"I like the little guy, he thinks like me"*, or *"It was my favorite. I started thinking 'Don't betray me, Yellow!' as it completed its sequence."*

Discussion

Our t-predictability planner worked as expected, with the t-predictable robots leading to the highest user prediction accuracy given the first t targets. Overall, we believe t-predictability will be important in a task for all ts, and hypothesize that optimizing for a weighted combination will perform best in practice.

We note that β is problem-specific and can be expected to decay as the difficulty of the task increases; in each setting, it can be estimated from participant data. Although β was chosen ahead of time in our experiment to be $\beta = 1$, our results are validated by the $r = 0.87$ correlation between expected and observed human error rates.

The optimal choice of t is also a subject for further investigation and is likely context-specific. Depending on the particular task, there should be a different trade-off between predictability of later actions and that of earlier actions.

6.3.4 Evaluation via In-Person Study

Having tested our ability to produce t-predictable sequences, we next ran an in person study to test their implications. Participants used a smartphone to operate a remote-controlled Sphero BB-8 robot, and had to predict and adapt to the actions of an autonomous Pioneer P3DX robot in a collaboration scenario (Figure 6.14).

Experiment Setup

Independent Variables We manipulated one single variable, *planner*, as a within-subjects factor. Having confirmed the expected effects of the different planners in the previous experiment, and given the good overall performance of the 1-predictable planner across different conditions, we decided to omit the 2-predictable agent and focus on testing the implications of 1-predictable with respect to optimal in a more immersive collaborative context.

Procedure At the beginning of the experiment, participants were told that they and their two robot friends were on a secret mission to deactivate an artifact. In each of 4 trials, the autonomous P3DX navigated to the 5 power sources and deactivated them in sequence; however, security sensors activated at each power source after 3 or more had been powered down. The subject's mission was to use BB-8 to jam the sensors at the third, fourth and fifth power sources before the P3DX arrived at them, by steering BB-8 into the corresponding sensor for a short period of time.

After an initial practice phase in which participants had a chance to familiarize themselves with the objective and rules of the task, as well as with the BB-8 teleoperation interface, there were two blocks of 4 trials whose order was counterbalanced across participants. In each block, the subject collaborated with the P3DX under a different task planner which we referred to as different robot "personalities".

Stimuli. Each of the 5 power sources (targets) in each trial was projected onto the floor as a yellow circle, using an overhead projector. Each circle was initially surrounded by a projected blue ring representing a dormant sensor. When the P3DX reached a target, both the circle and the ring were eliminated, except When the P3DX reached the third target, in which case the blue circles turned red symbolizing their switch into active state. Whenever BB-8 entered a ring, the ring turned green for 2 seconds and then disappeared, indicating successful jamming. If the P3DX moved over a red ring, a large red rectangle was projected, symbolizing capture and the trial ended in failure. Conversely, if the P3DX completed all 5 targets without entering a red ring, a green rectangle indicated successful completion of the trial.

Layout Generation. The 4 layouts used were taken from the larger pool of 15 layouts in the online experiment. There was a balance between layouts where online participants had been more accurate with the optimal planner, more accurate with the 1-predictable planner, or similarly accurate.

Controlling for Confounds. We controlled for confounds by counterbalancing the order of the planners; by using a practice layout; and by randomizing the trial order.

Dependent Measures

Objective measures. We recorded the number of successful trials for each subject and robot planner, as well as the number of trials where participants jammed targets in the correct sequence.

Subjective measures. After every block of the experiment, each participant was also asked to complete a questionnaire (Table 6.4) to evaluate their perceived performance of the P3DX robot. At the end of the experiment, we asked participants to indicate which robot (planner) they preferred working with.

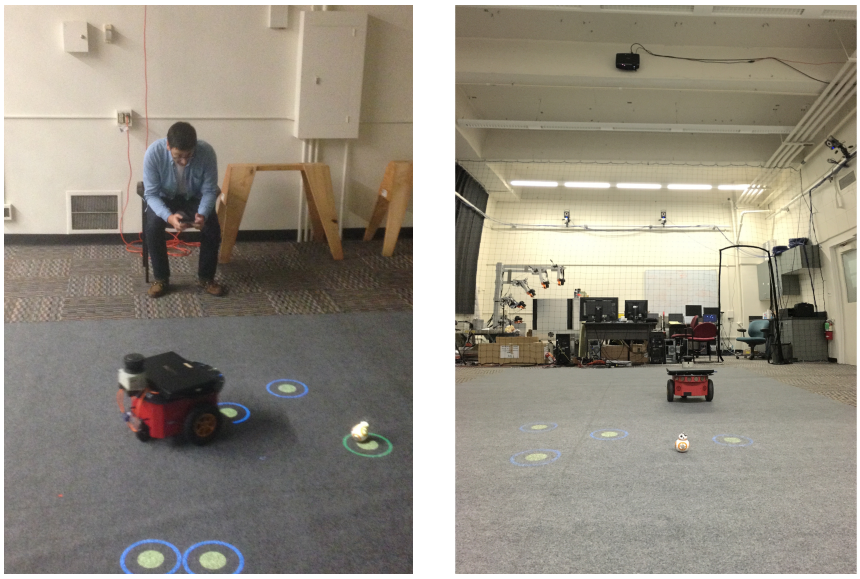


Figure 6.14: The experiment setup. Left: a participant is remotely controlling the Sphero BB-8 to clear the target that he believes will be third in the P3-DX's sequence. Right: the VICON system (infrared cameras) and the projector.

Hypotheses

H4 - Comparison with Optimal. *The* 1*-predictable robot will result in more successful trials than the optimal baseline.*

H5 - Preference. *Users will prefer working with the* 1*-predictable robot.*

Participants We recruited 14 participants from the UC Berkeley community, who were treated in accordance with local IRB standards and paid $10 USD. The study took about 30 min.

Results

Successful completions We first looked at how often participants were able to complete the task with each robot. We constructed a logistic mixed-effects model for completion success with planner type as a fixed effect and participant and task layout as random effects. We found a significant effect of planner type ($\chi^2(1) = 11.17, p < 0.001$), with the 1-predictable robot yielding more successful completions than the optimal robot ($z = 3.34,\ p < 0.001$). This supports **H4**.

Prediction accuracy We also looked at how accurate participants were at predicting the robots' sequence of tasks, based on the order in which participants jammed tasks. We constructed a logistic mixed-effects model for prediction accuracy with planner type as a fixed effect and participant and task layout as random effects. We found a significant effect of planner type ($\chi^2(1) = 9.49, p < 0.01$), with the 1-predictable robot being more predictable than the optimal robot ($z = 3.08,\ p < 0.01$).

Table 6.4: Subjective Measures

Legibility
1. The robot was moving in a way that helped me figure out what order it was planning to do the targets in.
2. The robot's initial actions made it clear what the rest of the plan was.
3. I found it hard to predict the last three targets in the right order.
4. Overall, I felt that the robot's choice of sequence made my prediction job easier.
Capability
1. The robot seemed to know what it was doing.
2. The robot chose its actions poorly.
Fluency
1. The robot and I worked fluently together.
2. I found it difficult to collaborate with the robot in a fluent way.
Trust
1. The robot was trustworthy.
2. I trusted the robot to do the right thing at the right time.
Predictability
1. I often found the robots overall sequence of targets confusing.
2. I found the robot's first target choice confusing.
3. The robot chose a reasonable sequence of targets.

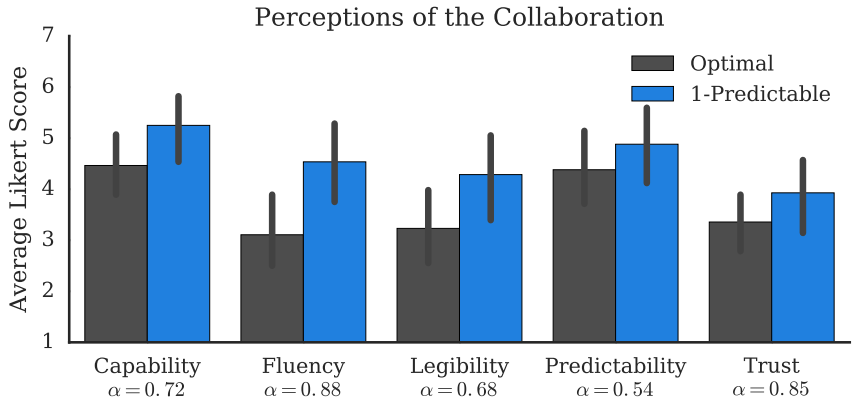


Figure 6.15: Perceptions of the collaboration. Over all measures, participants ranked the 1-predictable planner as being preferable to the optimal planner.

Robot preferences We asked participants to pick the robot they preferred to collaborate with. We found that 86% ($N = 12$) of participants preferred the predictable robot, while the rest ($N = 2$) preferred the optimal robot. This result is significantly different from chance ($\chi^2(1) = 7.14, p < 0.01$). This supports **H5**.

Perceptions of the collaboration We analyzed participants' perceptions of the robots' behavior (Fig. 6.15) by averaging each participant's responses to the individual questions for each robot and measure, resulting in a single score per participant, per measure, per robot. We constructed a linear mixed-effects model for the survey responses with planner and measure type as fixed effects, and with participants as random effects. We found a main effect of planner ($F(1, 117) = 16.42,\ p < 0.001$) and measure ($F(4, 117) = 5.45,\ p < 0.001$). Post-hoc comparisons using the multivariate t method for p-value adjustment indicated that participants preferred the predictable robot over the optimal robot ($t(117) = 4.05,\ p < 0.001$) by an average of 0.87 ± 0.21 SE points on the Likert scale.

Participant comments Based on participants' feedback, the BB-8 robot takes some efforts to control, which negatively affected their interaction with the P-3DX robot. However, they still noticed the difference between the two "personalities" of P-3DX. Some said when working with the optimal robot, "bb8 is a little hard to control within the time constraint" while in the legible robot, he/she commented that "I feel easier in the second set of trials". One participant also said that "the second robot seemed clever than the first one. Again, it was hard for me to direct the same robot. I was kind of more focused on the small robot than the big robot to get it go where I wanted". The comments support our hypothesis that the legible robot is easier and desirable to collaborate with. In our future in-person study, we will eliminate the negative effects introduced by the control of a robot to get a more "clean" experiment results.

Discussion

We conducted an in-person user study and the results show that our t-predictability planner worked as expected, with 1-predictable robots leading to the highest user prediction accuracy and the perceived collaboration. The comments from participants also shows people's preference over the 1-predictable robot. The user study further confirms our belief in the importance of t-predictability in human-robot collaboration.

6.4 Summary

The popularity of robots has led to increasing interest in research on human-robot collaboration. The misalignments between human's belief and the automation's actual plan have unfortunately caused serious issues. Works in this chapter have made initial efforts on improving human-robot collaboration by focusing on the inference that an agent makes. We propose a task allocation approach that combines motion-level intention inference with task-level planning. We use the Bayesian inference method for the robot to infer the human intention. The MAP estimate is then used for task allocation by using an MILP formulation. We tested the effectiveness of robot adaptation in improving the collaboration between human and robot in a large-scale online experiment, in which a human user interacted with three types of robots in a setting of multi-agent Traveling Salesman Problem. The experiment results show that by adding adaptivity into robot, both the objective team performance and the perceived robot's performance are improved.

We also propose the concept of t-predictability that quantitatively describes a plan in terms of the easiness that a human observer can accurately predict it after seeing only part of the plan. We then developed the theoretical formulation for computing action sequences, called the t-predictable planner, based on a Boltzmann model of human inference. The t-predictable planner enables a robot to generate a t-predictable plan that a human can confidently infer the rest of the plan after observing the first t actions. We tested the ability to make plans t-predictable in a large-scale online experiment in the setting of an open Traveling Salesman Problem, in which subjects' predictions of the robot's action sequence significantly improved, as well as the perceived robot's performance. The experiment results also show high correlation between the theoretical and the empirical error rates in plan prediction. In an in-person study, we investigated the effects of the t-predictability on human-robot collaboration, indicating that t-predictability can lead to significant objective and perceived improvements compared to traditional optimal planning. This study could prove beneficial for developing human-compatible robots in the future.

Chapter 7

Conclusion

The rapid progress of robotic systems has aroused growing interest and demands on making robots play important roles in the development of our society. As a consequence, robots have started entering human workplace and interacting with people. This raises new challenges to robotic systems. They are expected to not only be functionally capable, but also become human-compatible, which requires robots to become competent assistants to work for people and collaborative partners to work with people. This dissertation identifies three critical problems that robots are faced with, including the distributed filtering, informative path planning, and task planning for human-robot collaboration, and develops corresponding approaches for each of these problems. The first part of the dissertation, including Chapters 3 and 4, is focused on the estimation and planning of robotic systems, motivated by the practical needs of target localization and tracking using robots. The work in this part takes advantage of robots' superior capability in mobility, sensing, and planning to make robots effectively work for people on tasks that are laborious and difficult for human workers. The second part of the dissertation, including Chapters 5 and 6, is focused on the interaction between humans and robots. Chapter 5 studies the path planning of human-following robots under the uncertainty of human trajectory, while Chapter 6 investigates the task planning of robots that is explicitly purposed for improving robots' coordination with humans by focusing on the plan and inference that people make. The contributions in each chapter are summarized as follows:

Chapter 3 proposes distributed Bayesian filters for multi-agent systems. Instead of transmitting estimation statistics, each robot transmits sensor measurements to their neighboring robots. The following contributions enable an efficient distributed filter with performance guarantees:

- The work proposes the LIFO and FIFO methods for information sharing. For time-invariant communication networks, LIFO guarantees the intermittent dissemination of measurements among all robots. For time-variant networks, we propose the frequently jointly strongly connectedness network connectivity condition, under which FIFO ensures that each robot's sensor measurements can be transmitted to any other robots frequently enough.

- Two distributed Bayesian filters, LIFO-DBF and FIFO-DBF, are developed. The former is applicable to time-invariant communication networks while the latter is for time-variant networks. It has been shown that compared to current statistics-dissemination-based distributed filters, the proposed DBFs require less communication burden, which is desirable for current

multi-agent systems, in which each agent usually has limited communication bandwidth. Besides, the consistency of LIFO-DBF and FIFO-DBF are proved in the case of static target. The consistency guarantee makes the proposed DBFs advantageous to other distributed nonlinear filtering approaches that no consistency guarantee are obtained to the best of the author's knowledge.

- Simulation and experiment results have shown the effectiveness of proposed DBFs. Especially, DBFs can achieve accurate estimation that is comparable to a centralized filter, and are more superior than consensus-based filters.

Chapter 4 investigates the informative path planning of robots for target search and tracking in a unified framework using model predictive control. To deal with the issues including the complexity in computing the objective function, the requirement of collision avoidance, and the limited sensing domain, the work in this chapter makes the following contributions:

- This work develops a sequential path planning approach for the case of linear Gaussian sensing and mobility model. The sensing domain is geometrically modeled and the effects of limited FOV are incorporated in the planning process via the use of a modified Kalman filter that is designed for filtering under intermittent measurements.

- The work subsequently considers the situation where a binary sensor is used for localizing a static target in a dynamic environment that contains obstacles. The particle filter is used for target position estimation. We derive an analytical expression of the objective function by using GMM approximation of the target distribution, which significantly reduces the computational complexity of solving MPC. The barrier function is used in the objective function to ensure the safety of the robot.

Driven by the growing needs of using robots in shared workplace to work with humans, Chapter 5 embarks on the planning in the presence of people. This chapter focuses on the path planning of human-following robots under the uncertainty of human future trajectory and makes following contributions to make the robot a safe and comfortable companion:

- This work develops a model-based human motion prediction approach, namely PIMM-UKF, that can make short-term prediction of human trajectory. This method achieves more accurate prediction than traditionally used model-based prediction approaches and does not require training data, which makes it applicable to various environments.

- A path planning approach is developed using nonlinear model predictive control. The human spatial behavior theory and the "Proxemics" model are used for designing robot behavior to make the accompanied person feel comfortable. The safety is guaranteed via the collision avoidance constraints in the MPC formulation. Simulation results have shown the effectiveness of the proposed prediction and planning methods in generating safe and comfortable human-following path.

While Chapter 5 is focused on path planning, Chapter 6 investigates the task planning under the uncertainty of human intention and belief. This chapter makes the following contributions to improve human-robot collaboration:

- The work develops an adaptive task planning approach that uses motion-level goal inference to adapt the robot's task-level plan. The Bayesian intention inference theory is used for inferring the human's goal. The inference result is then utilized for task planning by solving a mixed integer linear program. A large-scale online experiment has validated the effectiveness of the proposed approach in improving human-robot collaboration both subjectively and objectively.
- The work subsequently considers generating human-predictable behaviors. We model the human's prediction using the Boltzmann policy, and based on this modeling, we propose the metric t-predictability, which evaluates a plan in terms of the easiness that a human observer can accurately predict it after seeing only part of the plan. We then develop a t-predictable planner for the robot to generate human-predictable plans. Both simulation and in-person study show that t-predictable plans can lead to significant objective and perceived improvements compared to traditional optimal planning.

7.1 Future Work

Works in Chapter 4 has made initial efforts in providing a systematic way of generating locally optimal plan under the non-Gaussian state uncertainty. A closely related and more general topic is the belief space planning, where both the states of the robot and the environment are uncertain. Belief space planning is an important topic in decision making theory. However, most works assume a Gaussian distribution of states. This dissertation work is promising to be extended to general non-Gaussian belief space planning to generate locally optimal plans. In conjunction with the developed DBFs in Chapter 3 for state estimation, we can also extend the proposed planning approach to distributed version for multi-agent systems.

In terms of human-robot collaboration, this dissertation work has considered the case that the robot is in an assistive role (Chapter 5 and Section 6.2), which assumes that the human is not affected by the robot's behavior, and considered the human's inference of the robot's plan but without any prediction of human's possible reaction (Section 6.3). Better collaboration results could conceivably be obtained if these limitations were to be relaxed and replaced by a more refined model of human inference and decision making, enabling a more effective adaptation of the robot's plan.

Appendix A

Appendix

A.1 Exponential Family of Distributions

In this section, we present the procedure to reformulating a multi-variate Gaussian distribution into the form of the Exponential Family. The Exponential Family of Distributions is defined as

$$g(x;\theta) = h(x)e^{\eta(\theta)^T T(x) - A(\theta)}, \tag{A.1}$$

where x and θ represent variables and parameters respectively. A d-dimensional Gaussian distribution with known mean μ and covariance Σ is defined as

$$q(x;\mu,\Sigma) = (2\pi)^{-\frac{d}{2}}|\Sigma|^{-\frac{1}{2}}e^{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)}.$$

The equation $q(x;\mu,\Sigma)$ can be written in the form of Eq. (A.1) with the following parameters:

$$\theta = [\mu, \Sigma], \tag{A.2a}$$

$$T(x) = \left[x, -xx^T\right]^T, \tag{A.2b}$$

$$\eta(\theta) = \left[\Sigma^{-1}\mu, \frac{1}{2}\Sigma^{-1}\right]^T, \tag{A.2c}$$

$$h(x) = 1, \tag{A.2d}$$

$$A(\theta) = \frac{1}{2}\mu^T\Sigma^{-1}\mu + \frac{1}{2}|\Sigma| + \frac{d}{2}ln(2\pi). \tag{A.2e}$$

Define

$$\Lambda = \Sigma^{-1}\mu, \tag{A.3a}$$

$$\Psi = \frac{1}{2}\Sigma^{-1}. \tag{A.3b}$$

Let $\Phi = [\Lambda, \Psi]$. Notice that Eq. (A.3a) and Eq. (A.3b) correspond to $\eta(\theta)$ in Eq. (A.2c). Then $q(x;\mu,\Sigma)$ can be reformulated using Φ as the parameter:

$$q(x;\Phi) = e^{\Phi^T T(x) - A(\Phi)}, \tag{A.4}$$

where

$$A(\Phi) = \frac{1}{4}\Lambda^T\Psi^{-1}\Lambda + \frac{d}{2}ln(2\pi) - \frac{1}{2}|2\Psi|.$$

A.2 Deriving the Analytical Form of the Objective Function

It is known that the product of two Gaussian probability density functions (PDFs) is a Gaussian function (not a Gaussian PDF). To be specific, let $f(x)$ and $g(x)$ be two Gaussian PDFs in the form of Eq. (A.4):

$$f(x) = e^{\Phi_1^T T(x) - A(\Phi_1)},$$
$$g(x) = e^{\Phi_2^T T(x) - A(\Phi_2)}.$$

Their product is

$$\begin{aligned} r(x) &= f(x)g(x) \\ &= e^{(\Phi_1+\Phi_2)^T T(x) - A(\Phi_1) - A(\Phi_2)} \\ &= e^{A(\Phi_1+\Phi_2) - A(\Phi_1) - A(\Phi_2)} e^{(\Phi_1+\Phi_2)^T T(x) - A(\Phi_1+\Phi_2)} \\ &= s e^{(\Phi_1+\Phi_2)^T T(x) - A(\Phi_1+\Phi_2)}. \end{aligned}$$

where $s = e^{A(\Phi_1+\Phi_2) - A(\Phi_1) - A(\Phi_2)}$ is a constant. Therefore $r(x)$ is a Gaussian function with the parameter $\Phi_1 + \Phi_2$. This closure property is applied to the non-detection function eq. (4.17a). For the purpose of simplicity, we derive the case of $H = 2$.

$$J(y_k^R, u_{k:k+1}^R) = \int_S \sum_{j=1}^{n} v_j e^{\Phi_{j,k}^T T(x^t) - A(\Phi_{j,k})} \prod_{i=1}^{2} \left[1 - c e^{{\Phi_{k+i}^R}^T T(x^t) - A(\Phi_{k+i}^R)}\right] \mathrm{d}x^t, \tag{A.6a}$$

$$\begin{aligned} &= \int_S \sum_{j=1}^{n} v_j e^{\Phi_{j,k}^T T(x^t) - A(\Phi_{j,k})} [1 - c e^{{\Phi_{k+1}^R}^T T(x^t) - A(\Phi_{k+1}^R)} - c e^{{\Phi_{k+2}^R}^T T(x^t) - A(\Phi_{k+2}^R)} \\ &+ c^2 e^{(\Phi_{k+1}^R + \Phi_{k+2}^R)^T T(x^t) - A(\Phi_{k+1}^R) - A(\Phi_{k+2}^R)}] \mathrm{d}x^t \\ &= \sum_{j=1}^{n} v_j \int_S e^{\Phi_{j,k}^T T(x^t) - A(\Phi_{j,k})} [1 - c e^{{\Phi_{k+1}^R}^T T(x^t) - A(\Phi_{k+1}^R)} - c e^{{\Phi_{k+2}^R}^T T(x^t) - A(\Phi_{k+2}^R)} \\ &+ c^2 e^{(\Phi_{k+1}^R + \Phi_{k+2}^R)^T T(x^t) - A(\Phi_{k+1}^R) - A(\Phi_{k+2}^R)}] \mathrm{d}x^t. \end{aligned} \tag{A.6b}$$

Consider the following integral Eq. (A.7) that appears as one part of the integral term associated with v_j in Eq. (A.6b). It can be simplified as follows:

$$\int_S e^{\Phi_{j,k}{}^T T(x^t) - A(\Phi_{j,k})} e^{\Phi^R_{k+1}{}^T T(x^t) - A(\Phi^R_{k+1})} \mathrm{d}x^t \tag{A.7}$$

$$= \int_S e^{(\Phi_{j,k} + \Phi^R_{k+1})^T T(x^t) - A(\Phi_{j,k}) - A(\Phi^R_{k+1})} \mathrm{d}x^t, \tag{A.8}$$

$$= e^{A(\Phi_{j,k} + \Phi^R_{k+1}) - A(\Phi_{j,k}) - A(\Phi^R_{k+1})} \int_S e^{(\Phi_{j,k} + \Phi^R_{k+1})^T T(x^t) - A(\Phi_{j,k} + \Phi^R_{k+1})} \mathrm{d}x^t, \tag{A.9}$$

$$= e^{A(\Phi_{j,k} + \Phi^R_{k+1}) - A(\Phi_{j,k}) - A(\Phi^R_{k+1})}. \tag{A.10}$$

Equation Eq. (A.10) is obtained from Eq. (A.9) since the exponential term to be integrated is a Gaussian probability density function, the integral of which equals 1. Define

$$\begin{aligned}
\alpha_{j,i_1} &= A(\Phi_{j,k} + \Phi^R_{k+1}) - A(\Phi_{j,k}) - A(\Phi^R_{k+1}), \\
\alpha_{j,i_2} &= A(\Phi_{j,k} + \Phi^R_{k+2}) - A(\Phi_{j,k}) - A(\Phi^R_{k+2}), \\
\alpha_{j,i_1,i_2} &= A(\Phi_{j,k} + \Phi^R_{k+1} + \Phi^R_{k+2}) - A(\Phi_{j,k}) - A(\Phi^R_{k+1}) - A(\Phi^R_{k+2}).
\end{aligned}$$

Then using similar method for deriving Eq. (A.10), Eq. (A.6b) can be simplified as

$$J(u_{k:k+1}) = 1 + \sum_{j=1}^{n} [-c v_j e^{\alpha_{j,i_1}} - c v_j e^{\alpha_{j,i_2}} + c^2 v_j e^{\alpha_{j,i_1,i_2}}].$$

The general formula of the non-detection function with horizon H is shown in Eq. (4.17b).

A.3 Approximating Obstacles Using Ellipses

Rectangular obstacles are approximated and analytically represented as ellipses. Let a and b be the length and width of a rectangular obstacle centered at the origin. Let Eq. (A.11) represent the ellipse that encloses the obstacle in the way that the four vertices of the rectangle lie on the boundary of the ellipse:

$$\frac{x^2}{\alpha^2} + \frac{y^2}{\beta^2} = 1. \tag{A.11}$$

In addition, assume that the rectangle and ellipse have the same aspect ratio, which means $\frac{a}{b} = \frac{\alpha}{\beta}$, then by simple algebraic manipulation, we can obtain that $\alpha = \frac{a}{\sqrt{2}}$ and $\beta = \frac{b}{\sqrt{2}}$. Define $h(x, y) = 2\frac{(x-c_1)^2}{a^2} + 2\frac{(y-c_2)^2}{b^2} - 1$. The $h(x, y) = 0$ represent the shifted-center ellipse approximation of the rectangle with length a and width b. Any point (x, y) with $h(x, y) > 0$ lies outside of the ellipse. Fig. A.1 shows the schematic plot of the approximating ellipse for the rectangular obstacle centered at (c_1, c_2).

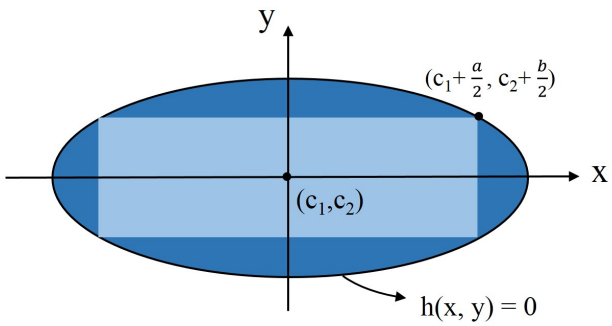


Figure A.1: Approximating the rectangle obstacle with an ellipse.

A.4 Task Allocation Using MILP

Assume we have n tasks and two agents, i.e., one human and one robot agent. Let $\mathcal{V} = \{1, 2\}$, with $v = 1$ referring to the human agent and $v = 2$ representing the robot. As discussed in Section 6.2, there are two types of tasks. The one-agent tasks are labeled as $1 \dots n_1$ and the joint tasks as $n_1 + 1 \dots n$. Therefore $\mathcal{G} = \{1, \dots, n\}$. The allocation result contains a set of binary decision variables in the form of $y^v_{i,j}$ and a set of continuous nonnegative variables a^v_k and d^v_k, $v \in \mathcal{V}$. $y^v_{i,j} = 1$ indicates that agent v travels from position i to position j directly and $y^v_{i,j} = 0$ indicates no movement from i to j. a^v_k and d^v_k represent the arrival and departure time of agent v at position k. The set of positions available as the from-states i in $y^v_{i,j}$ includes both task positions $\mathcal{G}$ and agents' current positions while the set of to-states j includes task positions $\mathcal{G}$ and a virtual "sink" state for each agent. When in the "sink" state, an agent stops moving and stays where it is afterwards. The set of positions k in a^v_k and d^v_k includes all the task positions and agent positions. For notational simplicity, let $\mathbf{y}_{alloc}$ denote the joint allocation variable that contains all the aforementioned binary variables $y^v_{i,j}$ and $\mathbf{t}_{alloc}$ represent the joint time variable that contains all the time variables a^v_k and d^v_k. For the case of w agents and n tasks, $\mathbf{y}_{alloc}$ is a binary-valued vector with $(n + w) \times (n + 1) \times w$ elements and $\mathbf{t}_{alloc}$ is a nonnegative vector with $2 \times (n + w) \times w$ elements. Specifically, for the problem considered in this work, $w = 2$. $j = n + 1$ refers to the "sink" state and $i = n + 1$, $i = n + 2$ represent the starting positions of the human and the robot.

The formulation of constraints in Eq. (6.2) are described below:

- Non-timing constraints:

  1. Each agent visits a set of one-agent tasks. All one-agent tasks need to be visited once:

$$\sum_{i=1, i\neq j}^{n+w} y^1_{i,j} + y^2_{i,j} = 1, \quad j = 1 \dots n_1.$$

2. Both agents visit each joint task once:

$$\begin{bmatrix} \sum_{i=1,i\neq j}^{n+w} y_{i,j}^{1} \\ \sum_{i=1,i\neq j}^{n+w} y_{i,j}^{2} \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad j = n_1 + 1 \ldots n.$$

3. An agent can leave a task position no more than once:

$$\sum_{j=1,j\neq i}^{n+1} y_{i,j}^{v} \leq 1, \quad i = 1 \ldots n, \; v = 1, 2.$$

4. An agent can leave a task position only after visiting it:

$$\sum_{j=1,j\neq i}^{n+1} y_{i,j}^{v} = \sum_{k=1}^{n+w} y_{k,i}^{v}, \quad i = 1 \ldots n, v = 1, 2.$$

5. No subtour exists for any agent:

$$\sum_{i,j\in\mathbf{S}} y_{i,j}^{v} \leq |\mathbf{S}| - 1, \quad v = 1, 2, \; \forall \mathbf{S} \subseteq \mathbf{M} = \{1, \ldots, n\} : |\mathbf{S}| \geq 2.$$

6. All agents leave their starting positions:

$$\sum_{j=1}^{n+1} y_{n+v,j}^{v} = 1, \quad v = 1, 2.$$

7. The inferred human task should be assigned to the human as her first task to complete:

$$y_{n+1,h_{inf}}^{1} = 1,$$

where h_{inf} represents the inferred human task.

- Timing constraints:

1. Both agents leave their starting positions at the beginning:

$$d_{n+v}^{v} = 0, \quad v = 1, 2.$$

2. An agent's arrival time at a corresponding one-agent task equals its departure time:

$$\begin{aligned} d_i^1 &= a_i^1, \quad i = 1 \ldots n, \\ d_i^2 &= a_i^2, \quad i = 1 \ldots n. \end{aligned}$$

3. An agent's arrival time at a joint task is no greater than its departure time:

$$d_i^v \geq a_i^v, \quad i = n_1 + 1 \ldots n, \ v = 1, 2.$$

4. The human departure time at a joint task equals the robot's if they are at the same joint task:

$$d_i^1 = d_i^2, \quad i = n_1 + 1 \ldots n.$$

5. If an agent moves from position i to j, then arrival time at j equals the departure time at i and the traveling time from i to j:

$$a_j^v = d_i^v + t_{i,j}^v, \quad v = 1, 2, \ i = 1 \ldots n, \ j = 1 \ldots n + w, \ \text{if } y_{i,j}^v = 1,$$

where $t_{i,j}^v$ denotes the time for the agent v to move from position i to j. Since the "if" statement does not fall into the MILP framework, we reformulate this equality constraint into two inequality constraints.

$$\begin{aligned} a_j^v + M \cdot (1 - y_{i,j}^v) &\geq d_i^v + t_{i,j}^v, \quad v = 1, 2, \ i = 1 \ldots n, \ j = 1 \ldots n + w, \\ a_j^v &\leq d_i^v + t_{i,j}^v + M \cdot (1 - y_{i,j}^v), \quad v = 1, 2, \ i = 1 \ldots n, \ j = 1 \ldots n + w, \end{aligned}$$

where M is a large number.

Bibliography

[1] Jake K Aggarwal and Quin Cai. "Human motion analysis: A review". In: *Computer vision and image understanding* 73.3 (1999), pp. 428–440.

[2] John R Aiello. "Human spatial behavior". In: *Handbook of environmental psychology* 1.1987 (1987), pp. 389–504.

[3] Rachid Alami, Aurelie Clodic, Vincent Montreuil, Emrah Akin Sisbot, and Raja Chatila. "Toward Human-Aware Robot Task Planning." In: *AAAI Spring Symposium: To Boldly Go Where No Human-Robot Team Has Gone Before*. 2006, pp. 39–46.

[4] Rachid Alami, Alin Albu-Schaffer, Antonio Bicchi, Rainer Bischoff, Raja Chatila, Alessandro De Luca, Alfredo De Santis, Georges Giralt, Jeremie Guiochet, Gerd Hirzinger, et al. "Safe and dependable physical human-robot interaction in anthropic domains: State of the art and challenges". In: *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*. IEEE. 2006, pp. 1–16.

[5] Takeshi Amemiya. *Advanced econometrics*. Harvard university press, 1985.

[6] Sean Andrist, Xiang Zhi Tan, Michael Gleicher, and Bilge Mutlu. "Conversational gaze aversion for humanlike robots". In: *Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction*. ACM. 2014, pp. 25–32.

[7] Michael Argyle. *Bodily communication*. Routledge, 2013.

[8] Nikolay A Atanasov, Jerome Le Ny, and George J Pappas. "Distributed algorithms for stochastic source seeking with mobile robot networks". In: *Journal of Dynamic Systems, Measurement, and Control* 137.3 (2015), p. 031004.

[9] Chris L Baker, Rebecca Saxe, and Joshua B Tenenbaum. "Action understanding as inverse planning". In: *Cognition* 113.3 (2009), pp. 329–349.

[10] Saptarshi Bandyopadhyay and Soon-Jo Chung. "Distributed estimation using bayesian consensus filtering". In: *American Control Conference (ACC), 2014*. IEEE. 2014, pp. 634–641.

[11] Craig Earl Beal and J Christian Gerdes. "Model predictive control for vehicle stabilization at the limits of handling". In: *IEEE Transactions on Control Systems Technology* 21.4 (2013), pp. 1258–1269.

[12] Jonathan Beaudeau, Monica F Bugallo, and Petar M Djuric. "Target tracking with asynchronous measurements by a network of distributed mobile agents". In: *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pp. 3857–3860.

[13] Michael Beetz, Freek Stulp, Piotr Esden-Tempski, Andreas Fedrizzi, Ulrich Klank, Ingo Kresse, Alexis Maldonado, and Federico Ruiz. "Generality and legibility in mobile manipulation". In: *Autonomous Robots* 28.1 (2010), pp. 21–44.

[14] Roger Bemelmans, Gert Jan Gelderblom, Pieter Jonker, and Luc De Witte. "Socially assistive robots in elderly care: a systematic review into effects and effectiveness". In: *Journal of the American Medical Directors Association* 13.2 (2012), pp. 114–120.

[15] Jeff A Bilmes. "A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models". In: *International Computer Science Institute* 4.510 (1998), p. 126.

[16] Adrian N Bishop, Baris Fidan, Brian DO Anderson, Kutluyil Dogancay, and Pubudu N Pathirana. "Optimality analysis of sensor-target localization geometries". In: *Automatica* 46.3 (2010), pp. 479–492.

[17] Devin Bonnie, Salvatore Candido, Timothy Bretl, and Seth Hutchinson. "Modelling search with a binary sensor utilizing self-conjugacy of the exponential family". In: *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pp. 3975–3982.

[18] Frederic Bourgault, Tomonari Furukawa, and Hugh F Durrant-Whyte. "Optimal search for a lost target in a Bayesian world". In: *Field and service robotics*. Springer. 2006, pp. 209–222.

[19] Frederic Bourgault, Tomonari Furukawa, and Hugh F Durrant-Whyte. "Process model, constraints, and the coordinated search strategy". In: *Robotics and Automation, 2004 IEEE International Conference on*. Vol. 5. IEEE. 2004, pp. 5256–5261.

[20] Misel Brezak and Ivan Petrovic. "Real-time approximation of clothoids with bounded error for path planning applications". In: *IEEE Transactions on Robotics* 30.2 (2014), pp. 507–515.

[21] D.C. Brogan and N.L. Johnson. "Realistic human walking paths". In: *11th IEEE International Workshop on Program Comprehension* (2003). ISSN: 1087-4844.

[22] Allison Bruce and Geoffrey Gordon. "Better motion prediction for people-tracking". In: *Proc. of the Int. Conf. on Robotics & Automation (ICRA), Barcelona, Spain*. 2004.

[23] Allison Bruce, Illah Nourbakhsh, and Reid Simmons. "The role of expressiveness and attention in human-robot interaction". In: *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*. Vol. 4. IEEE. 2002, pp. 4138–4142.

[24] Henry Carrillo, Philip Dames, Vijay Kumar, and Joss A Castellanos. "Autonomous robotic exploration using a utility function based on Renyi's general theory of entropy". In: *Autonomous Robots* (2017), pp. 1–22.

[25] Ashwin Carvalho, Yiqi Gao, Andrew Gray, H Eric Tseng, and Francesco Borrelli. "Predictive control of an autonomous ground vehicle using an iterative linearization approach". In: *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*. IEEE. 2013, pp. 2335–2340.

[26] Derek Stanley Caveney. *Multiple model techniques in automotive estimation and control*. Tech. rep. 2004.

[27] Jean-Francois Chamberland and Venugopal V Veeravalli. "Wireless sensors in distributed detection applications". In: *Signal Processing Magazine, IEEE* 24.3 (2007), pp. 16–25.

[28] Wesley P Chan, Iori Kumagai, Shunichi Nozawa, Yohei Kakiuchi, Kenichi Okada, and Masayuki Inaba. "Implementation of a robot-human object handover controller on a compliant underactuated hand using joint position error measurements for grip force and load force estimations". In: *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE. 2014, pp. 1190–1195.

[29] Eugene Charniak and Robert P Goldman. "A Bayesian model of plan recognition". In: *Artificial Intelligence* 64.1 (1993), pp. 53–79.

[30] Benjamin Charrow, Vijay Kumar, and Nathan Michael. "Approximate representations for multi-robot control policies that maximize mutual information". In: *Autonomous Robots* 37.4 (2014), pp. 383–400.

[31] Benjamin Charrow, Nathan Michael, and Vijay Kumar. "Cooperative multi-robot estimation and control for radio source localization". In: *The International Journal of Robotics Research* 33.4 (2014), pp. 569–580.

[32] Zhe Chen. "Bayesian filtering: From Kalman filters to particle filters, and beyond". In: *Statistics* 182.1 (2003), pp. 1–69.

[33] Timothy H Chung, Moshe Kress, and Johannes O Royset. "Probabilistic search optimization and mission assignment for heterogeneous autonomous agents". In: *Robotics and Automation, 2009. IEEE International Conference on*. IEEE. 2009, pp. 939–945.

[34] Mark Coates. "Distributed particle filters for sensor networks". In: *Proceedings of the 3rd international symposium on Information processing in sensor networks*. ACM. 2004, pp. 99–107.

[35] Akansel Cosgun, Dinei A Florencio, and Henrik I Christensen. "Autonomous person following for telepresence robots". In: *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE. 2013, pp. 4335–4342.

[36] Gergely Csibra and Gyorgy Gergely. "'Obsessed with goals': Functions and mechanisms of teleological interpretation of actions in humans". In: *Acta psychologica* 124.1 (2007), pp. 60–78.

[37] Alexander Cunningham, Manohar Paluri, and Frank Dellaert. "DDF-SAM: Fully distributed SLAM using constrained factor graphs". In: *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. IEEE. 2010, pp. 3025–3030.

[38] Narsingh Deo. *Graph theory with applications to engineering and computer science*. Courier Dover Publications, 2016.

[39] M Bernardine Dias, Balajee Kannan, Brett Browning, E Gil Jones, Brenna Argall, M Freddie Dias, Marc Zinck, Manuela M Veloso, and Anthony J Stentz. "Sliding autonomy for peer-to-peer human-robot teams". In: *10th Intelligent Conference on Intelligent Autonomous Systems (IAS 2008), Germany*. 2008, pp. 332–341.

[40] Petar M Djuric, Jonathan Beaudeau, and Monica F Bugallo. "Non-centralized target tracking with mobile agents". In: *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pp. 5928–5931.

[41] Petar M Djuric, Mahesh Vemula, and Monica F Bugallo. "Target tracking by particle filtering in binary sensor networks". In: *Signal Processing, IEEE Transactions on* 56.6 (2008), pp. 2229–2238.

[42] Anca D Dragan, Shira Bauman, Jodi Forlizzi, and Siddhartha S Srinivasa. "Effects of Robot Motion on Human-Robot Collaboration". In: *ACM/IEEE International Conference on Human-Robot Interaction (HRI)* (2015).

[43] Anca D. Dragan, Kenton C T Lee, and Siddhartha S. Srinivasa. "Legibility and predictability of robot motion". In: *ACM/IEEE International Conference on Human-Robot Interaction* (2013), pp. 301–308.

[44] Anca D Dragan and Siddhartha S Srinivasa. "A policy-blending formalism for shared control". In: *The International Journal of Robotics Research* 32.7 (2013), pp. 790–805.

[45] Anca Dragan and Siddhartha Srinivasa. "Formalizing Assistive Teleoperation". In: *Robotics, Science and Systems (RSS)* (2012).

[46] Anca Dragan and Siddhartha Srinivasa. "Integrating human observer inferences into robot motion planning". In: *Autonomous Robots* 37.4 (2014), pp. 351–368.

[47] Dave Ferguson, Thomas M Howard, and Maxim Likhachev. "Motion planning in urban environments". In: *Journal of Field Robotics* 25.11-12 (2008), pp. 939–960.

[48] Alan Fern, Sriraam Natarajan, Kshitij Judah, and Prasad Tadepalli. "A Decision-Theoretic Model of Assistance." In: *IJCAI*. 2007, pp. 1879–1884.

[49] Alan Fern, Sriraam Natarajan, Kshitij Judah, and Prasad Tadepalli. "A decision-theoretic model of assistance". In: *Journal of Artificial Intelligence Research* 50.1 (2014), pp. 71–104.

[50] Gonzalo Ferrer and Alberto Sanfeliu. "Behavior estimation for a complete framework for human motion prediction in crowded environments". In: *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE. 2014, pp. 5940–5945.

[51] Gonzalo Ferrer, Anais Garrell Zulueta, Fernando Herrero Cotarelo, and Alberto Sanfeliu. "Robot social-aware navigation framework to accompany people walking side-by-side". In: *Autonomous Robots* (2016), pp. 1–19.

[52] J Fisac, Chang Liu, Jessica B Hamrick, Shankar Sastry, J Karl Hedrick, Thomas L Griffiths, and Anca D Dragan. "Generating plans that predict themselves". In: *Workshop on the Algorithmic Foundations of Robotics (WAFR)*. 2016.

[53] Terrence Fong, Illah Nourbakhsh, Clayton Kunz, Lorenzo Fluckiger, John Schreiner, Robert Ambrose, Robert Burridge, Reid Simmons, Laura M Hiatt, Alan Schultz, et al. "The peer-to-peer human-robot interaction project". In: *AIAA Space*. Vol. 2005. 2005.

[54] Dieter Fox, Wolfram Burgard, and Sebastian Thrun. "Active markov localization for mobile robots". In: *Robotics and Autonomous Systems* 25.3-4 (1998), pp. 195–207.

[55] Chiara Fulgenzi, Christopher Tay, Anne Spalanzani, and Christian Laugier. "Probabilistic navigation in dynamic environment using rapidly-exploring random trees and gaussian processes". In: *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*. IEEE. 2008, pp. 1056–1062.

[56] Tomonari Furukawa, Frederic Bourgault, Benjamin Lavis, and Hugh F Durrant-Whyte. "Recursive Bayesian search-and-tracking using coordinated UAVs for lost targets". In: *IEEE International Conference on Robotics and Automation (ICRA)* (2006).

[57] Tomonari Furukawa, Lin Chi Mak, Hugh Durrant-Whyte, and Rajmohan Madhavan. "Autonomous bayesian search and tracking, and its experimental validation". In: *Advanced Robotics* 26.5-6 (2012), pp. 461–485.

[58] Mark Gales and Steve Young. "The application of hidden Markov models in speech recognition". In: *Foundations and trends in signal processing* 1.3 (2008), pp. 195–304.

[59] Yiqi Gao, Andrew Gray, Janick V Frasch, Theresa Lin, Eric Tseng, J Karl Hedrick, and Francesco Borrelli. "Spatial predictive control for agile semi-autonomous ground vehicles". In: *11th International Symposium on Advanced Vehicle Control*. 2012.

[60] Carlos E Garcia, David M Prett, and Manfred Morari. "Model predictive control: theory and practice-a survey". In: *Automatica* 25.3 (1989), pp. 335–348.

[61] Michael J Gielniak and Andrea L Thomaz. "Enhancing interaction through exaggerated motion synthesis". In: *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*. ACM. 2012, pp. 375–382.

[62] Michael J Gielniak and Andrea L Thomaz. "Generating anticipation in robot motion". In: *RO-MAN, 2011 IEEE*. IEEE. 2011, pp. 449–454.

[63] Erving Goffman. *The presentation of self in everyday life*. Harmondsworth, 1978.

[64] Michael Grant and Stephen Boyd. "Graph implementations for nonsmooth convex programs". In: *Recent Advances in Learning and Control*. Ed. by V. Blondel, S. Boyd, and H. Kimura. Lecture Notes in Control and Information Sciences. Springer-Verlag Limited, 2008, pp. 95–110.

[65] Ben Grocholsky, James Keller, Vijay Kumar, and George Pappas. "Cooperative air and ground surveillance". In: *Robotics & Automation Magazine, IEEE* 13.3 (2006), pp. 16–25.

[66] Dongbing Gu. "Distributed particle filter for target tracking". In: *Robotics and Automation, 2007 IEEE International Conference on*, pp. 3856–3861.

[67] T. M. Gureckis, J. Martin, J. McDonnell, R. S. Alexander, D. B. Markant, A. Coenen, J. B. Hamrick, and P. Chan. "psiTurk: An open-source framework for conducting replicable behavioral experiments online". In: *Behavioral Research Methods* (2015).

[68] Inc. Gurobi Optimization. *Gurobi Optimizer Reference Manual*. 2015. URL: `http://www.gurobi.com`.

[69] Edward T Hall, Ray L Birdwhistell, Bernhard Bock, Paul Bohannan, A Richard Diebold Jr, Marshall Durbin, Munro S Edmonson, JL Fischer, Dell Hymes, Solon T Kimball, et al. "Proxemics [and comments and replies]". In: *Current anthropology* (1968), pp. 83–108.

[70] Karol Hausman, Jorg Muller, Abishek Hariharan, Nora Ayanian, and Gaurav S Sukhatme. "Cooperative multi-robot control for target tracking with onboard sensing". In: *The International Journal of Robotics Research* 34.13 (2015), pp. 1660–1677.

[71] Simon Haykin. *Kalman filtering and neural networks*. Vol. 47. John Wiley & Sons, 2004.

[72] Ruijie He, Sam Prentice, and Nicholas Roy. "Planning in information space for a quadrotor helicopter in a GPS-denied environment". In: *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*. IEEE. 2008, pp. 1814–1820.

[73] J Karl Hedrick, Brandon Basso, Joshua Love, and Benjamin M Lavis. "Tools and techniques for mobile sensor network control". In: *Journal of Dynamic Systems, Measurement, and Control* 133.2 (2011), p. 024001.

[74] Michael Held and Richard M Karp. "The traveling-salesman problem and minimum spanning trees". In: *Operations Research* 18.6 (1970), pp. 1138–1162.

[75] Peter Henry, Christian Vollmer, Brian Ferris, and Dieter Fox. "Learning to navigate through crowded environments". In: *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE. 2010, pp. 981–986.

[76] Ondrej Hlinka, Franz Hlawatsch, and Petar M Djuric. "Distributed particle filtering in agent networks: A survey, classification, and comparison". In: *Signal Processing Magazine, IEEE* 30.1 (2013), pp. 61–81.

[77] Ondrej Hlinka, Ondrej Sluciak, Franz Hlawatsch, Petar M Djuric, and Markus Rupp. "Likelihood consensus and its application to distributed particle filtering". In: *Signal Processing, IEEE Transactions on* 60.8 (2012), pp. 4334–4349.

[78] Guy Hoffman. "Evaluating fluency in human-robot collaboration". In: *ACM/IEEE International Conference on Human-Robot Interaction (HRI), Workshop on Human Robot Collaboration* (2013).

[79] Guy Hoffman and Cynthia Breazeal. "Effects of anticipatory action on human-robot teamwork efficiency, fluency, and perception of team". In: *ACM/IEEE International Conference on Human-Robot Interaction (HRI)* (2007).

[80] Gabriel M Hoffmann and Claire J Tomlin. "Mobile sensor network control using mutual information methods and particle filters". In: *Automatic Control, IEEE Transactions on* 55.1 (2010), pp. 32–47.

[81] Geoffrey A Hollinger, Brendan Englot, Franz S Hover, Urbashi Mitra, and Gaurav S Sukhatme. "Active planning for underwater inspection and the benefit of adaptivity". In: *The International Journal of Robotics Research* 32.1 (2013), pp. 3–18.

[82] Geoffrey A Hollinger and Gaurav S Sukhatme. "Sampling-based robotic information gathering algorithms". In: *The International Journal of Robotics Research* 33.9 (2014), pp. 1271–1287.

[83] Sanghyun Hong, Tim Smith, Francesco Borrelli, and J Karl Hedrick. "Vehicle inertial parameter identification using extended and unscented Kalman filters". In: *Intelligent Transportation Systems-(ITSC), 2013 16th International IEEE Conference on*. IEEE. 2013, pp. 1436–1441.

[84] Ali Jadbabaie, Jie Lin, and A Stephen Morse. "Coordination of groups of mobile autonomous agents using nearest neighbor rules". In: *IEEE Transactions on automatic control* 48.6 (2003), pp. 988–1001.

[85] Ziya Jiang, Yuqing He, and Jianda Han. "Disturbance estimation for RUAV using UKF with acceleration measurement". In: *Mechatronics and Automation (ICMA), 2015 IEEE International Conference on*. IEEE. 2015, pp. 500–505.

[86] Brian J Julian, Michael Angermann, Mac Schwager, and Daniela Rus. "Distributed robotic sensor networks: An information-theoretic approach". In: *The International Journal of Robotics Research* 31.10 (2012), pp. 1134–1154.

[87] Simon J Julier and Jeffrey K Uhlmann. "Unscented filtering and nonlinear estimation". In: *Proceedings of the IEEE* 92.3 (2004), pp. 401–422.

[88] Thomas Kailath, Ali H Sayed, and Babak Hassibi. *Linear estimation*. Vol. 1. Prentice Hall Upper Saddle River, NJ, 2000.

[89] Rahul Kala and Kevin Warwick. "Multi-level planning for semi-autonomous vehicles in traffic scenarios based on separation maximization". In: *Journal of Intelligent & Robotic Systems* 72.3-4 (2013), pp. 559–590.

[90] Rudolph Emil Kalman. "A new approach to linear filtering and prediction problems". In: *Journal of basic Engineering* 82.1 (1960), pp. 35–45.

[91] Sertac Karaman and Emilio Frazzoli. "Incremental sampling-based algorithms for optimal motion planning". In: *Robotics Science and Systems VI* 104 (2010).

[92] Richard Kelley, Alireza Tavakkoli, Christopher King, Monica Nicolescu, Mircea Nicolescu, and George Bebis. "Understanding human intentions via hidden markov models in autonomous mobile robots". In: *Proceedings of the 3rd ACM/IEEE international conference on Human robot interaction*. ACM. 2008, pp. 367–374.

[93] Cory D Kidd, Will Taggart, and Sherry Turkle. "A sociable robot to encourage social interaction among the elderly". In: *IEEE International Conference on Robotics and Automation (ICRA)* (2006).

[94] Dieter Koller, Joseph Weber, and Jitendra Malik. "Robust multiple car tracking with occlusion reasoning". In: *European Conference on Computer Vision*. Springer. 1994, pp. 189–196.

[95] Chris Kreucher, Keith Kastella, and Alfred O Hero Iii. "Sensor management using an active sensing approach". In: *Signal Processing* 85.3 (2005), pp. 607–624.

[96] Thibault Kruse, Amit Kumar Pandey, Rachid Alami, and Alexandra Kirsch. "Human-aware robot navigation: A survey". In: *Robotics and Autonomous Systems* 61.12 (2013), pp. 1726–1743.

[97] James J Kuffner and Steven M LaValle. "RRT-connect: An efficient approach to single-query path planning". In: *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*. Vol. 2. IEEE. 2000, pp. 995–1001.

[98] Yoshiaki Kuwata, Justin Teo, Gaston Fiore, Sertac Karaman, Emilio Frazzoli, and Jonathan P How. "Real-time motion planning with applications to autonomous urban driving". In: *IEEE Transactions on Control Systems Technology* 17.5 (2009), pp. 1105–1118.

[99] Pablo Lanillos, Eva Besada-Portas, Gonzalo Pajares, and Jose J Ruz. "Minimum time search for lost targets using cross entropy optimization". In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2012, pp. 602–609.

[100] John Lasseter. "Principles of traditional animation applied to 3D computer animation". In: *ACM Siggraph Computer Graphics*. Vol. 21. 4. ACM. 1987, pp. 35–44.

[101] Jean-Claude Latombe. *Robot motion planning*. Vol. 124. Springer Science & Business Media, 2012.

[102] Steven M LaValle. *Planning algorithms*. Cambridge university press, 2006.

[103] Steven M LaValle and James J Kuffner. "Randomized kinodynamic planning". In: *The International Journal of Robotics Research* 20.5 (2001), pp. 378–400.

[104] Donghan Lee, Chang Liu, Yi-Wen Liao, and J Karl Hedrick. "Parallel Interacting Multiple Model-Based Human Motion Prediction for Motion Planning of Companion Robots". In: *IEEE Transactions on Automation Science and Engineering* 14.1 (2017), pp. 52–61.

[105] Sung-On Lee, Young-Jo Cho, Myung Hwang-Bo, Bum-Jae You, and Sang-Rok Oh. "A stable target-tracking control for unicycle mobile robots". In: *Intelligent Robots and Systems, 2000 IEEE/RSJ International Conference on*. Vol. 3. IEEE. 2000, pp. 1822–1827.

[106] Claus Lenz, Suraj Nair, Markus Rickert, Alois Knoll, Wolgang Rosel, Jurgen Gast, Alexander Bannat, and Frank Wallhoff. "Joint-action for humans and industrial robots for assembly tasks". In: *Robot and Human Interactive Communication, 2008. RO-MAN 2008. The 17th IEEE International Symposium on*. IEEE. 2008, pp. 130–135.

[107] Keith YK Leung, Timothy D Barfoot, and Hugh HT Liu. "Decentralized localization of sparsely-communicating robot networks: A centralized-equivalent approach". In: *IEEE Transactions on Robotics* 26.1 (2010), pp. 62–77.

[108] Daniel S Levine. "Information-rich path planning under general constraints using rapidly-exploring random trees". PhD thesis. Massachusetts Institute of Technology, 2010.

[109] Chang Liu, Jessica B Hamrick, Jaime F Fisac, Anca D Dragan, J Karl Hedrick, S Shankar Sastry, and Thomas L Griffiths. "Goal inference improves objective and perceived performance in human-robot collaboration". In: *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*. International Foundation for Autonomous Agents and Multiagent Systems. 2016, pp. 940–948.

[110] Chang Liu and J Karl Hedrick. "Cooperative Search Using A Human-UAV Team". In: *AIAA Infotech@ Aerospace*. 2016, p. 1653.

[111] Chang Liu and J Karl Hedrick. "Model predictive control-based target search and tracking using autonomous mobile robot with limited sensing domain". In: *American Control Conference (ACC), 2017*. IEEE. 2017, pp. 2937–2942.

[112] Chang Liu, Seungho Lee, Scott Varnhagen, and H Eric Tseng. "Path planning for autonomous vehicles using model predictive control". In: *Intelligent Vehicles Symposium (IV), 2017 IEEE*. IEEE. 2017, pp. 174–179.

[113] Chang Liu, Shengbo Eben Li, and J Karl Hedrick. "Measurement Dissemination-Based Distributed Bayesian Filter Using the Latest-In-and-Full-Out Exchange Protocol for Networked Unmanned Vehicles". In: *IEEE Transactions on Industrial Electronics* 64.11 (2017), pp. 8756–8766.

[114] Chang Liu, Shengbo Eben Li, and J Karl Hedrick. "Model Predictive Control-Based Probabilistic Search Method for Autonomous Ground Robot in a Dynamic Environment". In: *ASME 2015 Dynamic Systems and Control Conference*.

[115] Chang Liu, Shengbo Li, Diange Yang, and J Karl Hedrick. "Distributed Bayesian Filter using Measurement Dissemination for Multiple UGVs with Dynamically Changing Interaction Topologies". In: *Journal of Dynamic Systems, Measurement, and Control* (2017).

[116] Chang Liu, Shih-Yuan Liu, Elena L Carano, and J Karl Hedrick. "A Framework for Autonomous Vehicles With Goal Inference and Task Allocation Capabilities to Support Peer Collaboration With Human Agents". In: *ASME 2014 Dynamic Systems and Control Conference*. American Society of Mechanical Engineers. 2014, V002T30A005–V002T30A005.

[117] Peng Liu and Umit Ozguner. "Predictive control of a vehicle convoy considering lane change behavior of the preceding vehicle". In: *2015 American Control Conference (ACC)*. IEEE. 2015, pp. 4374–4379.

[118] R Duncan Luce. *Individual choice behavior: A theoretical analysis*. Courier Corporation, 2005.

[119] James N MacGregor and Yun Chu. "Human performance on the traveling salesman and related problems: A review". In: *The Journal of Problem Solving* 3.2 (2011), p. 2.

[120] David JC MacKay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

[121] Raj Madhavan, Kingsley Fregene, and Lynne E Parker. "Distributed cooperative outdoor multirobot localization and mapping". In: *Autonomous Robots* 17.1 (2004), pp. 23–39.

[122] Mateusz Malanowski and Krzysztof Kulpa. "Two methods for target localization in multistatic passive radar". In: *IEEE transactions on Aerospace and Electronic Systems* 48.1 (2012), pp. 572–580.

[123] Ross Mead and Maja J Mataric. "Autonomous human–robot proxemics: socially aware navigation based on interaction potential". In: *Autonomous Robots* (2016), pp. 1–13.

[124] Lauren M Miller, Yonatan Silverman, Malcolm A MacIver, and Todd D Murphey. "Ergodic Exploration of Distributed Information". In: *IEEE Transactions on Robotics* 32.1 (2015), pp. 36–52.

[125] Philip M Morse. "Bernard Osgood Koopman, 1900–1981". In: *Operations Research* 30.3 (1982), pp. 417–427.

[126] Bilge Mutlu, Fumitaka Yamaoka, Takayuki Kanda, Hiroshi Ishiguro, and Norihiro Hagita. "Nonverbal leakage in robots: communication of intentions through seemingly unintentional behavior". In: *Proceedings of the 4th ACM/IEEE international conference on Human robot interaction*. ACM. 2009, pp. 69–76.

[127] Ryo Nakahashi, Chris L Baker, and Joshua B Tenenbaum. "Modeling human understanding of complex intentional action with a Bayesian nonparametric subgoal model". In: *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)* (2016).

[128] Stefanos Nikolaidis, Ramya Ramakrishnan, Keren Gu, and Julie Shah. "Efficient model learning from joint-action demonstrations for human-robot collaborative tasks". In: *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction*. ACM. 2015, pp. 189–196.

[129] Stefanos Nikolaidis and Julie Shah. "Human-robot cross-training: Computational formulation, modeling and evaluation of a human team training strategy". In: *International Conference on Human-Robot Interaction* (2013), pp. 33–40.

[130] Reza Olfati-Saber. "Distributed Kalman filter with embedded consensus filters". In: *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05. 44th IEEE Conference on*, pp. 8179–8184.

[131] Reza Olfati-Saber, Alex Fax, and Richard M Murray. "Consensus and cooperation in networked multi-agent systems". In: *Proceedings of the IEEE* 95.1 (2007), pp. 215–233.

[132] Reza Olfati-Saber, Elisa Franco, Emilio Frazzoli, and Je S Shamma. "Belief consensus and distributed hypothesis testing in sensor networks". In: *Networked Embedded Sensing and Control*. Springer, 2006, pp. 169–182.

[133] Nikolaos P Papanikolopoulos, Pradeep K Khosla, and Takeo Kanade. "Visual tracking of a moving target by a camera mounted on a robot: A combination of control and vision". In: *ieee Transactions on Robotics and Automation* 9.1 (1993), pp. 14–35.

[134] Sachin Patil, Yan Duan, John Schulman, Ken Goldberg, and Pieter Abbeel. "Gaussian belief space planning with discontinuities in sensing domains". In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2014, pp. 6483–6490.

[135] Plamen Petrov and Fawzi Nashashibi. "Modeling and nonlinear adaptive control for autonomous vehicle overtaking". In: *IEEE Transactions on Intelligent Transportation Systems* 15.4 (2014), pp. 1643–1656.

[136] Giovanni Pezzulo, Francesco Donnarumma, and Haris Dindo. "Human sensorimotor communication: A theory of signaling in online social interactions". In: *PLoS One* 8.11 (2013), e79876.

[137] Aurelio Piazzi, CG Lo Bianco, Massimo Bertozzi, Alessandra Fascioli, and Alberto Broggi. "Quintic G 2-splines for the iterative steering of vision-based autonomous vehicles". In: *IEEE transactions on Intelligent Transportation Systems* 3.1 (2002), pp. 27–36.

[138] Anawat Pongpunwattana and Rolf Rysdyk. "Real-time planning for multiple autonomous vehicles in dynamic uncertain environments". In: *Journal of Aerospace Computing, Information, and Communication* 1.12 (2004), pp. 580–604.

[139] Joshue Perez Rastelli, Ray Lattarulo, and Fawzi Nashashibi. "Dynamic trajectory generation using continuous-curvature algorithms for door to door assistance vehicles". In: *2014 IEEE Intelligent Vehicles Symposium Proceedings*. IEEE. 2014, pp. 510–515.

[140] Wei Ren, Randal W Beard, et al. "Consensus seeking in multiagent systems under dynamically changing interaction topologies". In: *IEEE Transactions on automatic control* 50.5 (2005), pp. 655–661.

[141] Alejandro Ribeiro and Georgios B Giannakis. "Bandwidth-constrained distributed estimation for wireless sensor networks-part II: unknown probability density function". In: *Signal Processing, IEEE Transactions on* 54.7 (2006), pp. 2784–2796.

[142] Jorge Rios-Martinez, Anne Spalanzani, and Christian Laugier. "From proxemics theory to socially-aware navigation: a survey". In: *International Journal of Social Robotics* 7.2 (2015), pp. 137–153.

[143] Yong Rui and Yunqiang Chen. "Better proposal distributions: Object tracking using unscented particle filter". In: *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*. Vol. 2. IEEE. 2001, pp. II–786.

[144] Allison Ryan and J Karl Hedrick. "Particle filter based information-theoretic active sensing". In: *Robotics and Autonomous Systems* 58.5 (2010), pp. 574–584.

[145] Nadine B Sarter, David D Woods, and Charles E Billings. "Automation surprises". In: *Handbook of human factors and ergonomics* 2 (1997), pp. 1926–1943.

[146] Oliver C Schrempf, David Albrecht, and Uwe D Hanebeck. "Tractable probabilistic models for intention recognition based on expert knowledge". In: *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*. IEEE. 2007, pp. 1429–1434.

[147] John Schulman, Yan Duan, Jonathan Ho, Alex Lee, Ibrahim Awwal, Henry Bradlow, Jia Pan, Sachin Patil, Ken Goldberg, and Pieter Abbeel. "Motion planning with sequential convex optimization and convex collision checking". In: *The International Journal of Robotics Research* 33.9 (2014), pp. 1251–1270.

[148] Mac Schwager, Philip Dames, Daniela Rus, and Vijay Kumar. "A multi-robot control policy for information gathering in the presence of unknown hazards". In: *Proceedings of International Symposium on Robotics Research, Aug*. 2011.

[149] Julie A Shah, Patrick R Conrad, and Brian Charles Williams. "Fast Distributed Multi-agent Plan Execution with Dynamic Task Assignment and Scheduling." In: *ICAPS*. 2009.

[150] Julie Shah and Cynthia Breazeal. "An empirical analysis of team coordination behaviors and action planning with application to human–robot teaming". In: *Human Factors* 52.2 (2010), pp. 234–245.

[151] Xiaohong Sheng, Y-H Hu, and Parameswaran Ramanathan. "Distributed particle filter with GMM approximation for multiple targets localization and tracking in wireless sensor network". In: *Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium on*. IEEE. 2005, pp. 181–188.

[152] Bruno Sinopoli, Luca Schenato, Massimo Franceschetti, Kameshwar Poolla, Michael I Jordan, and Shankar S Sastry. "Kalman filtering with intermittent observations". In: *IEEE transactions on Automatic Control* 49.9 (2004), pp. 1453–1464.

[153] Lawrence D Stone. "OR Forum-What's Happened in Search Theory Since the 1975 Lanchester Prize?" In: *Operations Research* 37.3 (1989), pp. 501–506.

[154] Mikael Svenstrup, Thomas Bak, and Hans Jorgen Andersen. "Trajectory planning for robots in dynamic human environments". In: *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. IEEE. 2010, pp. 4293–4298.

[155] Daniel Szafir, Bilge Mutlu, and Terrence Fong. "Communication of intent in assistive free flyers". In: *Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction*. ACM. 2014, pp. 358–365.

[156] Leila Takayama, Doug Dooley, and Wendy Ju. "Expressing thought: improving robot readability with animation principles". In: *Proceedings of the 6th international conference on Human-robot interaction*. ACM. 2011, pp. 69–76.

[157] Frank Thomas, Ollie Johnston, and Frank. Thomas. *The illusion of life: Disney animation*. Hyperion New York, 1995.

[158] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic robotics*. MIT press, 2005.

[159] John Tisdale, ZuWhan Kim, and J Karl Hedrick. "Autonomous UAV path planning and estimation". In: *Robotics & Automation Magazine, IEEE* 16.2 (2009), pp. 35–42.

[160] Michael Tomasello, Malinda Carpenter, Josep Call, Tanya Behne, and Henrike Moll. "Understanding and sharing intentions: the origins of cultural cognition." In: *The Behavioral and brain sciences* 28.5 (2005), 675–91, discussion 691–735.

[161] Pete Trautman, Jeremy Ma, Richard M Murray, and Andreas Krause. "Robot navigation in dense human crowds: Statistical models and experimental studies of human–robot cooperation". In: *The International Journal of Robotics Research* 34.3 (2015), pp. 335–356.

[162] Dariusz Ucinski. "Optimal sensor location for parameter estimation of distributed processes". In: *International Journal of Control* 73.13 (2000), pp. 1235–1248.

[163] Vaibhav V Unhelkar, Ho Chit Siu, and Julie A Shah. "Comparative performance of human and mobile robotic assistants in collaborative fetch-and-deliver tasks". In: *Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction*. ACM. 2014, pp. 82–89.

[164] Chris Urmson, Joshua Anhalt, Drew Bagnell, Christopher Baker, Robert Bittner, MN Clark, John Dolan, Dave Duggins, Tugrul Galatali, Chris Geyer, et al. "Autonomous driving in urban environments: Boss and the urban challenge". In: *Journal of Field Robotics* 25.8 (2008), pp. 425–466.

[165] Edo S Van der Poort, Marek Libura, Gerard Sierksma, and Jack AA van der Veen. "Solving the k-best traveling salesman problem". In: *Computers & operations research* 26.4 (1999).

[166] Venugopal V Veeravalli and Pramod K Varshney. "Distributed inference in wireless sensor networks". In: *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* 370.1958 (2012), pp. 100–117.

[167] Mahesh Vemula, Monica F Bugallo, and Petar M Djuric. "Target tracking in a two-tiered hierarchical sensor network". In: *Acoustics, Speech and Signal Processing, 2006. IEEE International Conference on*. Vol. 4, pp. IV–IV.

[168] Cordula Vesper, Stephen Butterfill, Gunther Knoblich, and Natalie Sebanz. "A minimal architecture for joint action". In: *Neural Networks* 23.8 (2010), pp. 998–1003.

[169] R. Bhushan Gopaluni Vinay A. Bavdekar and Sirish L. Shah. "Evaluation of Adaptive Extended Kalman Filter Algorithms for State Estimation in Presence of Model-Plant Mismatch". In: *Preprints of the 10th IFAC International Symposium on Dynamics and Control of Process Systems*. IFAC. 2013, pp. 184–189.

[170] Andreas Wachter and Lorenz T Biegler. "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming". In: *Mathematical programming* 106.1 (2006), pp. 25–57.

[171] Kazuyoshi Wada and Takanori Shibata. "Living with seal robots – its sociopsychological and physiological influences on the elderly at a care house". In: *IEEE Transactions on Robotics* 23.5 (2007), pp. 972–980.

[172] Eric Wan, Ronell Van Der Merwe, et al. "The unscented Kalman filter for nonlinear estimation". In: *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*. IEEE. 2000, pp. 153–158.

[173] Hongchuan Wei and Silvia Ferrari. "A geometric transversals approach to sensor motion planning for tracking maneuvering targets". In: *IEEE Transactions on Automatic Control* 60.10 (2015), pp. 2773–2778.

[174] Kenton Williams and Cynthia Breazeal. "A reasoning architecture for human-robot joint tasks using physics-, social-, and capability-based logic". In: *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE. 2012, pp. 664–671.

[175] Yuanqing Xia, Jizong Shang, Jie Chen, and Guo-Ping Liu. "Networked data fusion with packet losses and variable delays". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 39.5 (2009), pp. 1107–1120.

[176] Feng Xiao and Long Wang. "Asynchronous consensus in continuous-time multi-agent systems with switching topology and time-varying delays". In: *Automatic Control, IEEE Transactions on* 53.8 (2008), pp. 1804–1816.

[177] Shuang Xiao, Zhan Wang, and John Folkesson. "Unsupervised robot learning to predict person motion". In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2015, pp. 691–696.

[178] Bar-Shalom Yaakov, XR Li, and Kirubarajan Thiagalingam. "Estimation with applications to tracking and navigation". In: *New York: Johh Wiley and Sons* 245 (2001).

[179] John Yen, Xiaocong Fan, Shuang Sun, Timothy Hanratty, and John Dumer. "Agents with shared mental models for enhancing team decision makings". In: *Decision Support Systems* 41.3 (2006), pp. 634–653.

[180] Shuangyan Yi, Zhenyu He, Xinge You, and Yiu-Ming Cheung. "Single object tracking via robust combination of particle filter and sparse representation". In: *Signal Processing* 110 (2015), pp. 178–187.

[181] Jiaying Yu, Shengbo Eben Li, Chang Liu, and Bo Cheng. "Dynamical tracking of surrounding objects for road vehicles using linearly-arrayed ultrasonic sensors". In: *2016 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2016, pp. 72–77.

[182] Guoxian Zhang, Silvia Ferrari, and M Qian. "An information roadmap method for robotic sensor path planning". In: *Journal of Intelligent & Robotic Systems* 56.1 (2009), pp. 69–98.

[183] Yu Zhang, Sarath Sreedharan, Anagha Kulkarni, Tathagata Chakraborti, Hankz Hankui Zhuo, and Subbarao Kambhampati. "Plan Explicability for Robot Task Planning". In: *RSS Workshop on Planning for Human-Robot Interaction: Shared Autonomy and Collaborative Robotics* (2016).

[184] Brian D Ziebart, Nathan Ratliff, Garratt Gallagher, Christoph Mertz, Kevin Peterson, J Andrew Bagnell, Martial Hebert, Anind K Dey, and Siddhartha Srinivasa. "Planning-based prediction for pedestrians". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2009).

[185] Julius Ziegler, Philipp Bender, Thao Dang, and Christoph Stiller. "Trajectory planning for Bertha-a local, continuous method". In: *2014 IEEE Intelligent Vehicles Symposium Proceedings*. IEEE. 2014, pp. 450–457.

[186] Long Zuo, Kishan Mehrotra, Pramod K Varshney, and Chilukuri K Mohan. "Bandwidth-efficient target tracking in distributed sensor networks using particle filters". In: *Information Fusion, 2006 9th International Conference on*, pp. 1–4.